NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report 32-1430

Calculation of Scattered Patterns From Asymmetrical Reflectors

Arthur C. Ludwig

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

February 15, 1970

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report 32-1430

Calculation of Scattered Patterns From Asymmetrical Reflectors

Arthur C. Ludwig

JET PROPULSION LABORATORY

CALIFORNIA INSTITUTE OF TECHNOLOGY

PASADENA, CALIFORNIA

February 15, 1970

Prepared Under Contract No. NAS 7-100
National Aeronautics and Space Administration

Preface

The work described in this report was performed by the Telecommunications Division of the Jet Propulsion Laboratory.

Acknowledgment

This report is based upon a dissertation presented to the faculty of the Graduate School, University of Southern California, in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Electrical Engineering).

The author wishes to acknowledge the encouragement and support of Zohrab Kaprielian, guidance committee chairman; Charles Weber and W. V. T. Rusch, dissertation committee co-chairmen; and Alan Schumitzky and Lloyd Welch, dissertation committee members. Also, it should be noted that W. V. T. Rusch did much of the original work on numerical evaluation of scattered patterns using the physical-optics technique upon which the author's research is based.

The author also wishes to acknowledge the assistance of Gerald Levy and Dan Bathker of JPL and, in particular, Phil Potter, whose work on spherical-wave theory provided the background for the author's research in that area.

Contents

Tables

Printouts

Contents (contd)

Printouts (contd)

Figures

Contents (contd)

Figures (contd)

Abstract

A computer program for evaluating the electromagnetic field pattern of a known arbitrary incident field scattered from a perfectly conducting reflector of arbitrary shape is presented.

It is shown that the commonly used assumption of far-field behavior for the incident field leads to poor results in some cases; an exact method is developed in which a spherical-wave expansion is used to represent the incident field, and it is shown that this method yields the correct result in these cases.

The scattering surface is represented by a Fourier type of expansion in one variable, where the coefficients are specified as tabular functions of the remaining variable. A tilted figure of revolution is a case of practical importance, and it is shown that, for small tilt angles, three Fourier components are usually sufficient to represent the surface.

A new nonlinear integration technique is used that is four to eight times faster than Simpson's rule integration, under specified conditions. Other techniques to maximize program efficiency are described, and a comparison is given that indicates the program to be 11 to 19 times faster than a similar existing program.

The subreflector and main reflector patterns of an 85-ft-diam Cassegrainian antenna are computed, using both the far-field approximation and a spherical-wave expansion to represent the incident fields. The results are compared, and it is shown that the spherical-wave expansion of the subreflector pattern agrees in remarkable detail with an experimentally measured pattern.

Exemplifying the general capabilities of the program, the patterns of a paraboloid with a displaced feed are computed for three cases, and the results are compared with approximate theory.

Calculation of Scattered Patterns From Asymmetrical Reflectors

I. Introduction

The physical-optics technique has been used with great success to compute the scattering from perfectly conducting surfaces, such as the subreflector of a Cassegrainian antenna (Ref. 1). In fact, the technique has yielded such accurate results that computed patterns have begun to replace a good deal of experimental data in antenna development and analysis (Refs. 2 and 3).

A major limitation of the physical-optics technique is that it requires the evaluation of a double integral with a rapidly oscillating integrand that typically results in the need for large blocks of stored data and increased computer time. In some cases, with special symmetry conditions, one integration may be performed analytically (Ref. 4). The program described in this report was developed so that the broad class of scattering problems, which do not necessarily have any symmetries, could be resolved in the most efficient manner possible. The great majority of computer time and storage requirements may be directly attributed to the numerical evaluation of the integral; therefore, the major contribution towards efficiency in this program is the application of a new integration technique (Ref. 5).

The case of scattering from an infinite plane reflector is one of the few electromagnetic scattering problems that has a simple known solution; the resulting field is, of course, identically the reflection of the incident field. An earlier version of the computer program described herein implicitly assumed far-field behavior for the incident-field pattern. When the program was checked, with the plane reflector problem as a test case, it was found that the computed scattered pattern was not the same as the incident pattern and that the result was dependent upon the wavelength of the incident field. (Data for this test case are presented in detail in Section III.)

Of the several possible explanations for this anomaly, it was hypothesized that the far-field behavior of the incident fields was the cause. A technique based upon spherical-wave theory (which yields an exact representation of the incident fields) was developed, and the application of this technique eliminated the error.

The spherical-wave representation of the incident fields is in principle completely general. Also, a general representation of the scattering surface was used in the

development of the program. Practical considerations required that some of this generality be sacrificed in the actual programming, but the theory could easily be applied to extend the results if required.

The only important assumption made in the development of this program was the physical-optics approximation in which the currents on the reflecting surface were assumed to be determined by the incident magnetic field. The validity of this procedure is discussed in Section II; however, the agreement between experimental and computed results presented in Section VI and elsewhere (see Refs. 1–3) is, in fact, the strongest demonstration that the physical-optics approximation is a useful and valid engineering method.

II. The Physical-Optics Technique

If the currents $\mathbf{K}$ induced on a surface S by incident fields $\mathbf{E}_i$ and $\mathbf{H}_i$ can be determined, then the scattered fields $\mathbf{E}_s$ and $\mathbf{H}_s$ caused by these currents are given by

$$\mathbf{E}_s(P) = -\frac{j}{4\pi\omega\epsilon}\int_S [(\mathbf{K}\cdot\nabla)\nabla + k^2\mathbf{K}]\frac{e^{-jkr}}{r}\,dS \tag{1a}$$

and

$$\mathbf{H}_s(P) = \frac{1}{4\pi}\int_S (\mathbf{K}\times\nabla)\frac{e^{-jkr}}{r}\,dS \tag{1b}$$

The coordinate system is shown in Fig. 1.

Since these equations (Ref. 6) are exact, the scattering problem consists of (1) determining the currents $\mathbf{K}$ and (2) evaluating the integrals.

One method for obtaining the currents is to use the following boundary condition on S:

$$\mathbf{n}\times(\mathbf{E}_i + \mathbf{E}_s) = 0 \tag{2}$$

This method leads to the following integral equation for the currents $\mathbf{K}$:

$$-\mathbf{n}\times\mathbf{E}_i = -\mathbf{n}\times\frac{j}{4\pi\omega\epsilon}\int_S [(\mathbf{K}\cdot\nabla)\nabla + k^2\mathbf{K}]\frac{e^{-jkr}}{r}\,dS \tag{3}$$

This formulation has been known for some time (Ref. 6), but the speed and size of the present generation of computers have made the technique much more attractive. This integral equation method has been used recently with great success in solving two-dimensional scattering problems and problems involving a surface of revolution (Ref. 7). The major disadvantage of this approach is that, in these restricted cases, the surface must not be wider than approximately 20 wavelengths (Refs. 8 and 9). In the case of an arbitrary surface, the limitation is more severe.

An approximate alternate method for obtaining the currents is to assume that, on areas directly illuminated by the source, the currents are the same as they would be if the incident fields were reflected optically:

$$\mathbf{K} = 2\mathbf{n}\times\mathbf{H}_i \tag{4}$$

On shadowed regions, the currents are assumed to be zero. These assumptions are known as the physical-optics approximation.

For smooth reflectors, this approximation is valid in the limit of zero wavelength (in Ref. 10, Cullen gives proof of this for the case of a convex body illuminated by a plane wave) and, in contrast to the integral-equation method, the approximation improves as surface dimensions become large compared to a wavelength.

However, not only does this approximation yield results when it is not practical to apply the integral-equation technique, but it yields the results almost free in terms of computer time. Therefore, the generally time-consuming integral-equation technique should be used only when the physical-optics approximation is inadequate. In the typical scattering problem, the currents are of little direct interest; they are only an intermediate step toward obtaining the scattered fields. Therefore, the physical-optics approximation should not be judged by whether it yields the correct currents, but by whether the final computed scattered fields are correct.

In some cases Eq. (4) is a poor approximation of the true currents, but it results in an excellent approximation of the true scattered fields. The reason for this may be understood by the following: let the true currents be $\mathbf{K}$, and define $\Delta\mathbf{K}$ by

$$\Delta\mathbf{K} \equiv \mathbf{K} - 2\,\mathbf{n}\times\mathbf{H}_i \tag{5}$$

Typically, $\Delta\mathbf{K}$ exhibits oscillatory behavior over the surface, and its net integrated contribution to the scattered fields is small. Watson (Ref. 11) has shown this analytically for the particular case of the fields in the focal region of a paraboloid that is large with respect to a

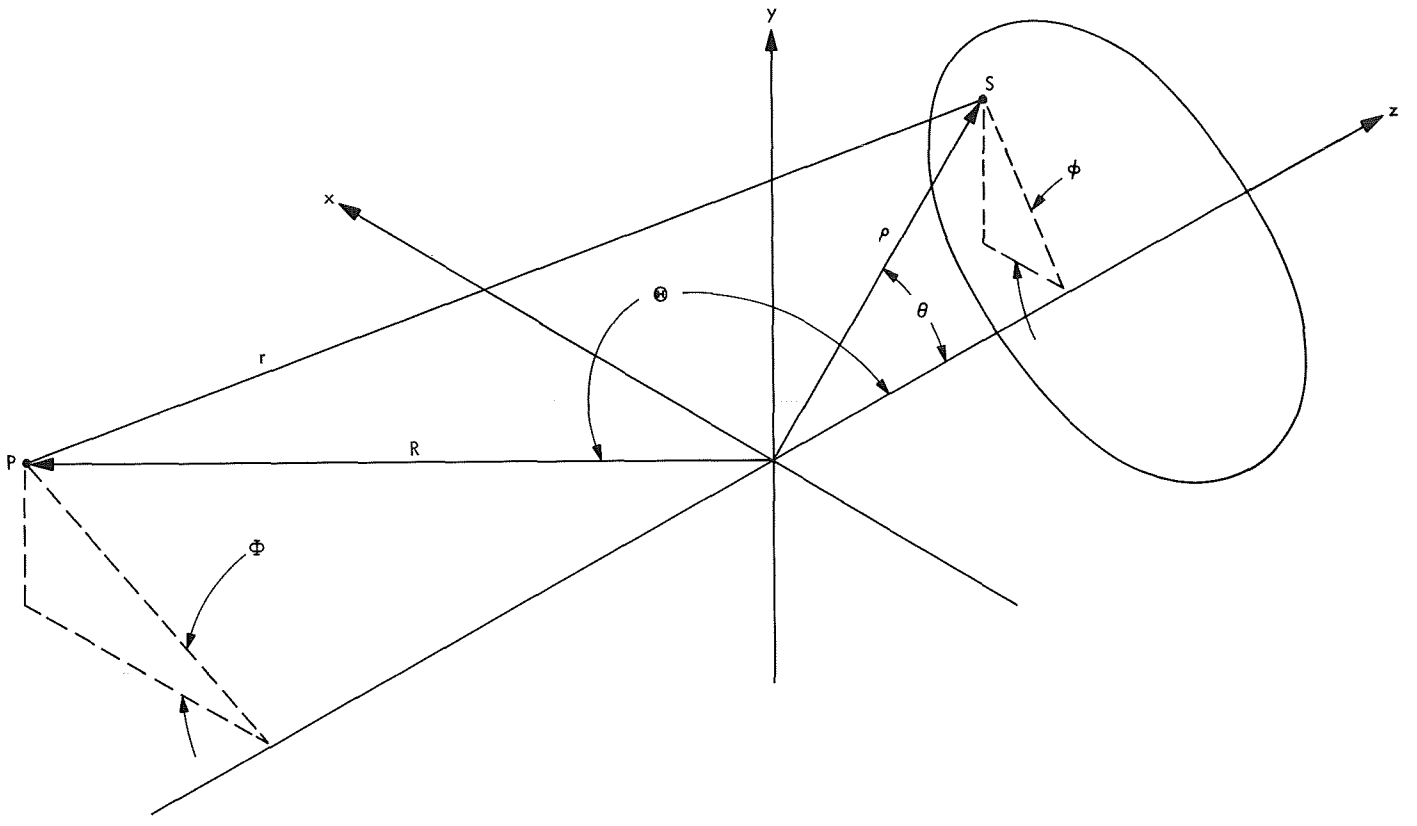


Fig. 1. Coordinate system

wavelength. Presumably, the primary effect of $\Delta\mathbf{K}$ is to produce local fields that represent stored rather than propagating energy.

The similarity between the currents computed by the integral equation technique and the physical-optics approximation has also been pointed out by Andreasen. The oscillatory nature of the difference may be seen in his results that are for scattering surfaces only a few wavelengths wide (Ref. 9). Surface currents for small spheres—0.18 to 3.2 wavelengths radius—obtained from a classical boundary value solution are given by King and Wu in Ref. 12, and even here, Eq. (4) is a reasonable approximation.

Therefore, Eq. (4) is not only a *necessary* step toward reaching a result in problems involving a large reflector, but also a sound engineering approximation for surfaces larger than a few wavelengths in size.

If it is assumed that $\mathbf{n}$ and $\mathbf{H}_i$ are known, the integrals of Eq. (1) can be evaluated to obtain the scattered fields at any point. However, the problem is greatly reduced if it is assumed that $\mathbf{E}_s$ and $\mathbf{H}_s$ are to be evaluated only at very large distances from S. As $R \to \infty$, $\mathbf{E}_s$ and $\mathbf{H}_s$ must satisfy the far-field relations:

$$\mathbf{H} = \left(\frac{\epsilon}{\mu}\right)^{1/2} \mathbf{i}_R \times \mathbf{E} \tag{6a}$$

and

$$\mathbf{E}(R,\Theta,\Phi) = [E_\Theta(\Theta,\Phi)\,\mathbf{i}_\Theta + E_\Phi(\Theta,\Phi)\,\mathbf{i}_\Phi]\,\frac{e^{-jkR}}{R} \tag{6b}$$

Therefore, only E_Θ and E_Φ need be evaluated. Also in this case, Eq. (1) may be written so that the only term in the integrand that is dependent on the location of the field point P is the argument of the exponential term. The remainder of the integrand may be initially computed over S and stored to be used repeatedly in the evaluation of the integral for every field point P. This method represents a substantial reduction in computer time.

The assumption that the scattered fields are to be evaluated at $R = \infty$ does not represent a loss of generality. If near-field results are required, they may be

obtained from the far-field data using the spherical-wave expansions described later.

The far-field version of Eq. (1), with the currents that are obtained from Eq. (14), is derived in Silver (Ref. 6):

$$\mathbf{E}_s(P) = I_\Theta \mathbf{i}_\Theta + I_\Phi \mathbf{i}_\Phi) \frac{e^{-jkR}}{R} \tag{7}$$

where

$$\mathbf{I} = -\frac{j\omega\mu}{2\pi} \int_S (\mathbf{n} \times \mathbf{H}_i) \exp(jk\boldsymbol{\rho} \cdot \mathbf{i}_R)\, dS$$

The final result, the total field $\mathbf{E}_T$, is obtained by adding the incident fields to the scattered fields as shown in Eq. (8):

$$\mathbf{E}_T = \mathbf{E}_S + \mathbf{E}_i \tag{8}$$

The problem has now been reduced to the following steps: (1) specify the surface data ρ and $\mathbf{n}$, (2) specify the incident field data $\mathbf{H}_i$, and (3) evaluate the integral.

III. Specification of the Scattering Surface

In the derivation of the scattering integral equations (see Eq. 1), the surface S is assumed to completely enclose some region of space. Examples of surfaces are a sphere (which separates space into an infinite region and a finite region) and an infinite plane (which separates space into two infinite regions). There are obvious numerical difficulties with infinite surfaces; the finite closed surface also presents some potential problems which are discussed in the following paragraphs.

As discussed in Section IV, the origin of the coordinate system is determined by the location of the source, and the natural system in which to describe the incident fields is spherical coordinates. To define a surface, it is necessary and sufficient to specify one variable as a function of the other two. The most natural and convenient choice for this problem is a function $\rho(\theta,\phi)$. In the case of a sphere that does not enclose the origin, a line $\theta = \text{const}$ $\phi = \text{const}$, in general, intersects the surface twice so it cannot be described by a single function of this form. However, with the assumed current approximation, the back portion of the sphere has zero currents and can be ignored. The illuminated portion is precisely the portion that can be defined by a single function. There is still one potential difficulty. At the point at which $\boldsymbol{\rho}$ is just tangent to the surface, the partial derivative $\partial\rho/\partial\theta$ is infinite and, as will be seen shortly, is required. Therefore, it is necessary to discard a further portion of the surface. In the case of an infinite surface, all but some finite portion is ignored. In both cases the remainder may be identified as a truncated surface. Therefore, an "arbitrary reflecting surface" is defined as a surface that can be represented by a function $\rho(\theta,\phi)$ whose partial derivatives exist at every point on the surface. This definition is basically a consequence of the physical-optics approximation used to obtain the currents, and is sufficiently general to include virtually all antenna reflectors. (Discontinuous surfaces may be represented by two or more segments satisfying these requirements, and the results may be superimposed.)

Because this surface may be thought of as a portion of a closed surface, the derivation of Eq. (1) is still valid, except for one point. A truncated surface has an edge, and the currents obtained from Eq. (14) will in general be discontinuous there. To ensure that the results are consistent with Maxwell's equations and the radiation condition, Silver (see Ref. 6) introduced a line charge on this edge in the derivation of Eq. (7). Sancer (Ref. 13) has shown that the surface integrals of Eq. (1) intrinsically contain the contour integral representing the contribution of this charge, and that the term introduced by Silver arises automatically as a mathematical consequence of truncating the surface.

The surface data required to evaluate the integral of Eq. (7) are ρ and $\mathbf{n}$. Using a result from differential geometry (Ref. 14), Dickinson (Ref. 15) pointed out the useful relationship:

$$\mathbf{n}dS = \frac{\partial\boldsymbol{\rho}}{\partial\phi} \times \frac{\partial\boldsymbol{\rho}}{\partial\theta}\, d\theta d\phi \tag{9}$$

where

$$\left.\begin{aligned} \boldsymbol{\rho} &= \rho\, \mathbf{i}_\rho \\ \frac{\partial\boldsymbol{\rho}}{\partial\phi} &= \frac{\partial\rho}{\partial\theta}\mathbf{i}_\rho + \rho \sin\theta\, \mathbf{i}_\phi \\ \frac{\partial\boldsymbol{\rho}}{\partial\theta} &= \frac{\partial\rho}{\partial\theta}\mathbf{i}_\rho + \rho\mathbf{i}_\theta \end{aligned}\right\} \tag{10}$$

Therefore, specification of ρ and its partial derivatives is a complete description of the surface. Since these quantities must be specified on a set of points on S, which will be called the integration grid and can easily involve

several thousand points, the storage and transmission of a description of a surface in this form can be a problem. For this reason, it is useful to achieve some sort of data compression. One trivial case is that in which the surface is completely described by an analytic expression; for example, a paraboloid whose focus is at the origin, is completely described by a single number f as shown in Eq. (11):

$$\rho = f \sec^2 \frac{\theta}{2} \tag{11}$$

A compromise between the complete generality of two-dimensional tabular input and the severe restriction of a simple analytic equation may be developed from the general form:

$$\rho(\theta,\phi) = \sum_{m=0}^{M} a_m(\theta) \cos m\phi + b_m(\theta) \sin m\phi \tag{12}$$

Then

$$\frac{\partial \rho}{\partial \theta} = \sum_{m=0}^{M} \frac{\partial a_m}{\partial \theta} \cos m\phi + \frac{\partial b_m}{\partial \theta} \sin m\phi \tag{13a}$$

and

$$\frac{\partial \rho}{\partial \phi} = \sum_{m=1}^{M} - m a_m(\theta) \sin m\phi + m b_m(\theta) \cos m\phi \tag{13b}$$

Suppose that for a given surface it is necessary to include terms of order up to M to determine ρ with sufficient accuracy; then by the sampling theorem it would be necessary to specify ρ at $2M$ values of ϕ to accurately represent details of the surface variations using tabular data. Therefore, this Fourier type of representation is in principle always at least as efficient as an ordinary tabular representation.

If M is allowed to be arbitrarily large, Eq. (12) is in fact completely general; however, the reason for choosing this form is that the first few terms are sufficient to accurately represent a slightly tilted surface of revolution, which is a case of substantial practical importance. For example, Fig. 2 shows the peak error that occurs when a tilted plane reflector is represented by three terms of the expansion, $a_0(\theta)$, $a_1(\theta)$, and $a_2(\theta)$. (The axis of rotation is taken to be the y-axis, so all of the $b_m(\theta)$ terms are zero.) This error is in general a function of the curvature of the reflector, but the data shown in the figure are

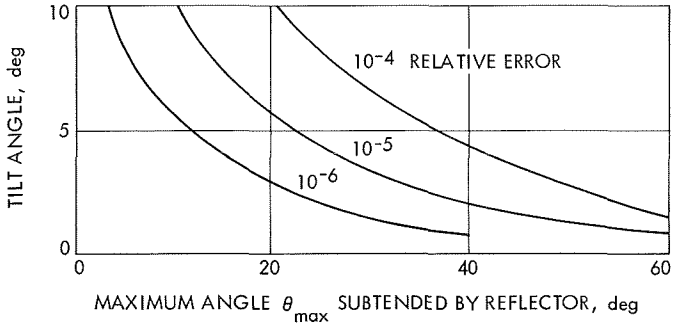


Fig. 2. Peak error with three Fourier components used to represent a tilted reflector

representative of the accuracy that may be obtained with this method. A reasonable *a posteriori* estimate of accuracy may be obtained by extrapolating the value of the first neglected term in the expansion. For example, for a tilt angle of 1 deg, the coefficients at $\theta_{max} = 20$ deg are $a_0 = 1.064199$, $a_1 = 0.006761$, and $a_2 = 0.000021$. The values decrease with a nearly constant ratio, and a reasonable estimate for the neglected third-order term is about 10^{-5} to 10^{-6}, which is approximately the peak error. The relatively poor accuracy obtained for a tilt angle of 10 deg at $\theta_{max} = 40$ deg is also predictable from the coefficients ($a_0 = 1.320015$, $a_1 = 0.197465$, and $a_2 = 0.014608$) and would lead to an expected relative error of approximately 10^{-3}.

The derivatives of the $a_m(\theta)$ terms must also be written in as tabular functions of θ; therefore, six input functions are required in this example where $M=3$ and all $b_m(\theta)$ terms are identically zero. The number of ϕ values at which ρ must be known is determined by the requirements of the numerical integration technique and is usually quite large (from fifty to several hundred values is typical). Therefore, the specification of only six functions is a substantial reduction in the quantity of data required. A computer program to obtain these functions for the case of an arbitrary tilt and translation of a surface of revolution is described in Appendix C.

IV. Specification of the Incident Fields

The most common method (Refs. 1, 6, and 16) for obtaining the magnetic field on the surface S is to assume that the incident fields satisfy the far-field relations:

$$\mathbf{H}_i = \left(\frac{\epsilon}{\mu}\right)^{1/2} \mathbf{i}_\rho \times \mathbf{E}_i \tag{14a}$$

and

$$\mathbf{E}_i(\rho, \theta, \phi) = [E_\theta(\theta, \phi)\,\mathbf{i}_\theta + E_\phi(\theta, \phi)\,\mathbf{i}_\phi]\,\frac{e^{-jk\rho}}{\rho} \qquad (14b)$$

Not only is this a relatively simple form, but $E_\theta(\theta, \phi)$ and $E_\phi(\theta, \phi)$ correspond to the quantities conventionally measured on an antenna pattern range, so experimental data may be easily used as input for the incident fields. Also, as discussed in Section II, theoretical patterns are easier to compute in this form.

Equation (14) represents a spherical wave diverging from the origin; therefore, it is not valid unless the origin is located at the source phase center. The phase pattern about this point should be constant, but in practice the relative phase may vary by 90 deg or more over the region of interest. A nonconstant phase pattern may be taken into consideration if E_θ and E_ϕ are allowed to be complex valued, but the radial components of the field implied by the nonconstant phase pattern are usually assumed negligible.

The use of this assumption provides good results for many practical problems. Serious errors can arise, however, when the surface is "close" to the source. This fact has been recognized by Zucker and Ierley, who directly computed the near-field radiation of a conical horn in evaluating the scattered pattern of a near-field Cassegrainian subreflector (Ref. 3). An alternative approach is to take experimental data at range values corresponding to the location of the surface; this was done by Hogg and Semplak (Ref. 17).

The disadvantage of the first method is that horn patterns computed in this way frequently do not agree with experimental data, since it is necessary that strong assumptions be made about the fields in the aperture of the horn. The disadvantage of the second method is that each different subreflector configuration in principle requires a new set of experimental data. These problems are obviated by the use of the spherical-wave expansions that are discussed later in this section.

A graphic demonstration of the error that may be introduced if far-field behavior is assumed for the incident fields is provided by the case of scattering from a large plane reflector. In the case of an infinite plane, the result must be the reflection of the incident pattern, independent of frequency. Also, in the case of an infinite plane, Eq. (4) is valid, independent of frequency; therefore, this is an excellent direct test of Eq. (14).

The cases shown in Fig. 3 (and one additional case at $z = 160\ \lambda$, which is not illustrated) were evaluated using the far-field form for the incident fields given by Eq. (14). The fields were derived from the experimental pattern of a conical horn with an aperture diameter of 4.671 λ, illustrated at the origin of Fig. 3. The actual numerical integration was truncated at an angle at which contributions became negligible. The disks shown in Fig. 3 represent the region over which the integral was actually evaluated. It should be noted that horn-reflector interactions are *not* considered, so this is not intended to be a perfect duplication of a real experimental situation, but rather a numerical test of the far-field assumption given by Eq. (14).

The computed scattered fields based on the far-field approximation of the incident field for the six cases are

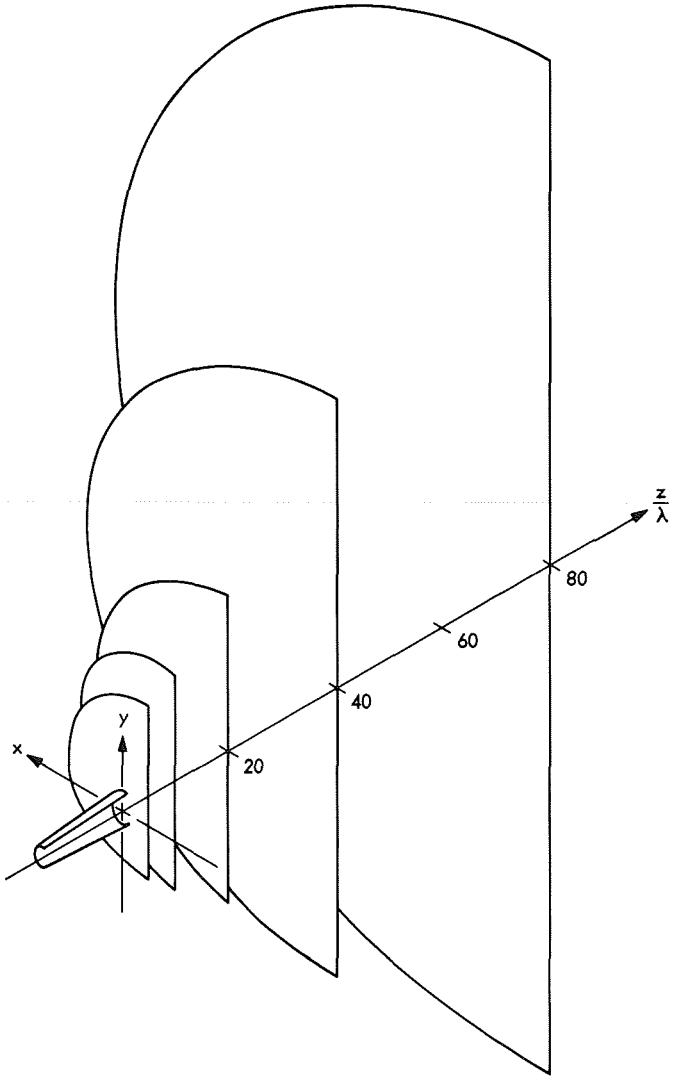


Fig. 3. Plane reflector cases

shown in Fig. 4 along with the incident-field pattern. Only the Φ-component is illustrated, but the behavior of the Θ-component is similar. All of these patterns should be identical, but the discrepancies are large. In addition to the severe distortion of the reflected main lobe when $z = 5$ and 10 λ, a spurious back lobe appears in all cases —that is, the plane that was assumed to be a perfect reflector behaves as if it were partially transparent.

In the special case of a plane reflector, the scattered fields $\mathbf{E}_s$ are identical in the reflected and backward directions. When the incident field is added (Eq. 8), it should cancel this backward radiation. Therefore, this back lobe is identically the error between the reflected pattern and the incident pattern. (The error includes both amplitude and phase differences since it is the difference between complex-valued quantities.)

The power in the computed scattered patterns has been calculated as a percentage of the power in the incident fields and is shown in Fig. 5. As indicated in Fig. 5, the main (reflected) lobe alone accounts for 100 percent of the incident power in all cases, so the total power in the computed pattern exceeds the incident power. This excess is, of course, physically impossible. The spurious power in the back lobe is strongly dependent on the distance

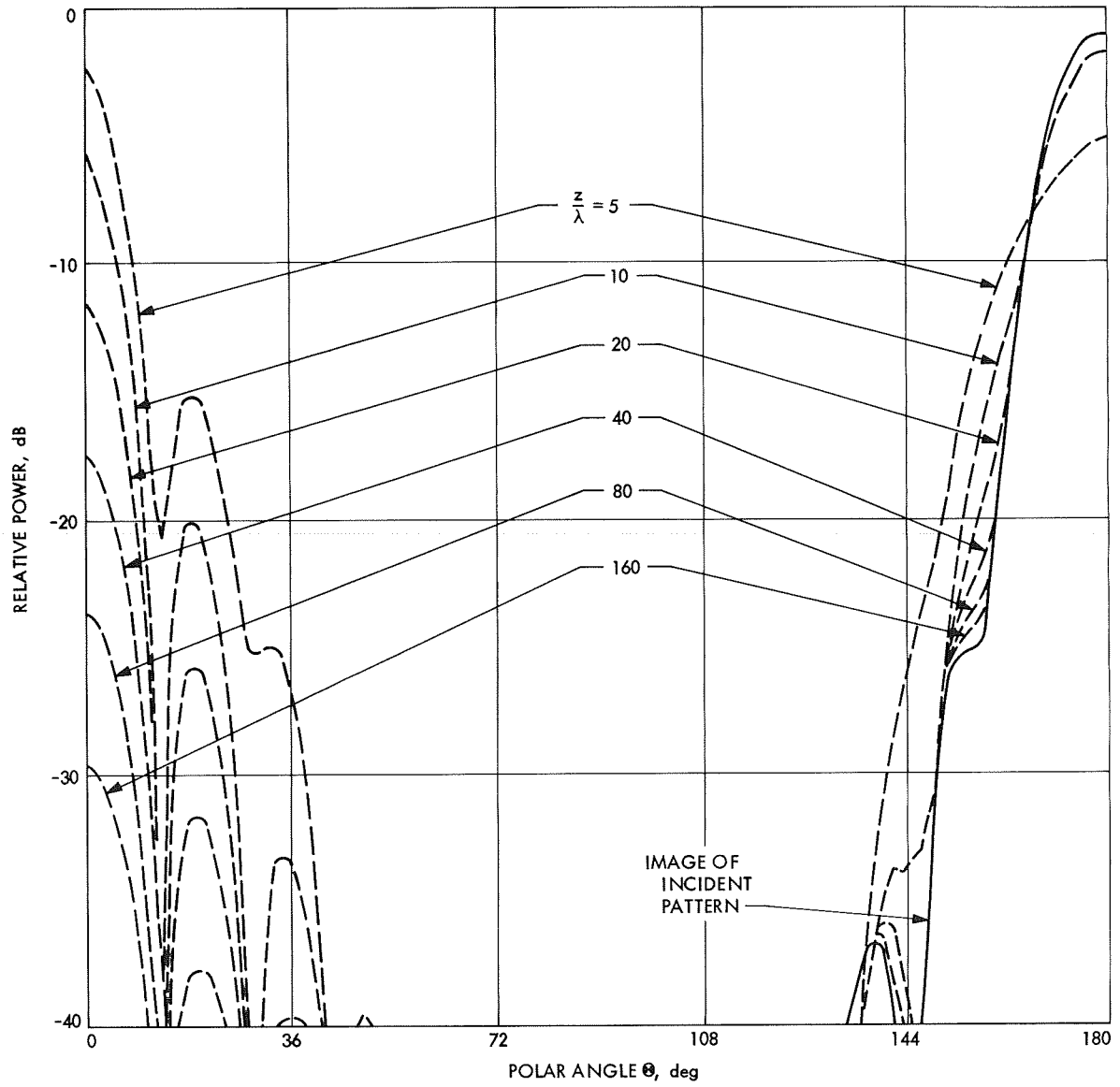


Fig. 4. Computed amplitude patterns for plane reflector cases; far-field assumption for incident fields

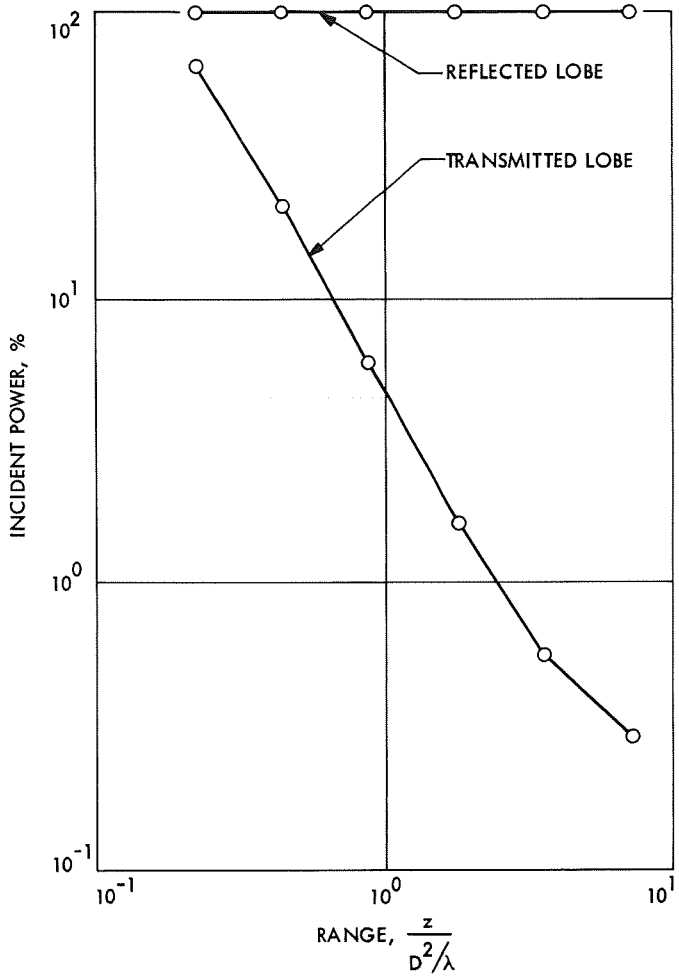


Fig. 5. Power in computed scattered patterns

from the source to the surface, which is plotted in units of D^2/λ where D is the diameter of the source. At $z=2D^2/\lambda$, the traditional far-field boundary line, this power has dropped to less than 2 percent.

The behavior of the phase pattern of E_θ is shown in Fig. 6. These phase data have been transformed to a new coordinate system; the origin was moved from the point at the source phase center to the image of this point behind the plane reflector where the phase center of the total scattered fields should be located. In fact, this is very nearly the phase center for the first three cases; the patterns of the last cases are those of a source whose phase center is closer to the plane surface.

Therefore, instead of the total scattered fields appearing to arise from a perfect image of the source, in the three worst cases the apparent source is distorted and has moved toward the plane. In most practical situations, the agreement shown for the three best cases would be adequate, but it is significant that even at the furthest distance $z = 160\,\lambda\,(7D^2/\lambda$ of the feed) there is noticeable error in the results.

This error may be eliminated entirely by the use of a representation of the incident fields that is valid in the near field. Spherical waves are a well known set of solutions to Maxwell's equations that satisfy this requirement. If the incident fields satisfy Maxwell's equations, then coefficients a_n and b_n may be found such that

$$\mathbf{H}_i\,(\rho, \theta, \phi) = \frac{k}{j\omega\mu} \sum_{n=0}^{\infty} a_n \mathbf{n}_n + b_n \mathbf{m}_n \qquad (15)$$

where $\mathbf{n}_n$ is a transverse electric (TE) and $\mathbf{m}_n$ is a transverse magnetic (TM) spherical wave, as given by Eq. (A-1) in Appendix A.

The method for obtaining coefficients so that the wave expansion will match an arbitrary input pattern in the far field is also discussed in Appendix A; therefore, it will simply be assumed that coefficients have been obtained

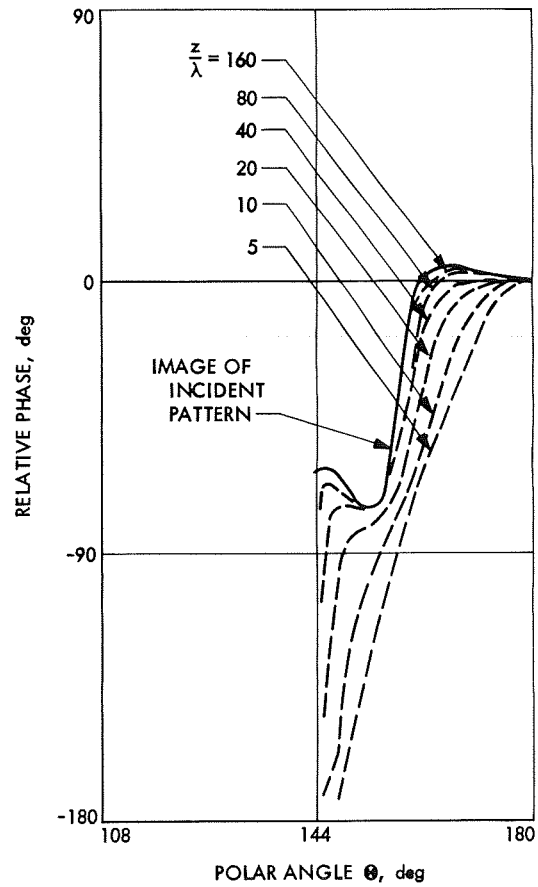


Fig. 6. Computed phase patterns for plane reflector cases; far-field assumption for incident fields

to describe the incident field. The form of Eq. (15) completely eliminates the approximation of Eq. (14) by directly yielding $\mathbf{H}_i$ on the surface. However, the evaluation of the spherical-wave expansion does involve computer time. Although this time is typically less than 10 percent of the total, the comment made earlier with regard to the integral-equation technique for obtaining the currents also applies here: the expansion should be used only when the far-field form is inadequate.

Clearly, the far-field form is inadequate in cases $z = 5$, 10, and 20 λ of the data presented in Figs. 4 and 6. These cases have been recomputed using a spherical-wave coefficient specification of the incident fields. The reflected patterns are virtually indistinguishable from the incident pattern, as shown in Fig. 7. Phase data, transformed to the image point, are shown in Fig. 8; clearly, the error has been eliminated.

Now that it has been established that the far-field assumption yields incorrect results that are corrected when a spherical-wave expansion is used, it is of interest to consider the physical mechanism of the discrepancy. Since the theory is based on current integration, the surface current distribution must reflect the essential difference between the two methods. The amplitude and phase of the y component (the principal polarization) of the surface currents at $\phi = 0$ are compared for three cases in Figs. 9 and 10. The currents in Figs. 9 and 10 are plotted versus the Cartesian coordinate x, rather than the polar angle θ (see Figs. 1 and 3), to illustrate the true relative spatial distributions of the three cases.

When the far-field approximation is assumed, the diameter of the region containing significant current magnitudes linearly approaches zero as the source approaches the reflector. Typically, a smaller source will result in a broader radiation pattern, but in this case the phase of the surface currents becomes more uniform as the source becomes smaller and this fact tends to make the pattern narrower. Over a considerable range of z values, these effects nearly cancel out, as demonstrated by the resulting radiation patterns shown in Fig. 4. However, in the limiting case of a point source, a broad dipole pattern would be expected with the phase center of the pattern on the reflecting surface. This is in fact exactly the trend exhibited by the data for the far-field case in Figs. 4 and 6. As mentioned previously, the surface currents radiate symmetrically on both sides of the plane, and because a broader pattern cannot result in perfect destructive interference with the incident pattern, the result is the back lobe which appears in Fig. 4.

When the spherical-wave expansion is used, the phase behavior is very similar to the far-field result, except that the 180-deg discontinuities are smoothed out. (Since the phase data are modulo 360 deg, these discontinuities could be in either the positive or negative direction. They were drawn to match the spherical-wave phase data as closely as possible.) The major difference between the spherical-wave and far-field cases is that the patterns of the current magnitudes are broadened, particularly in the $z = 5$ λ case. Because the phase patterns are nearly the same, the larger source produces a narrower radiation pattern that brings the result into agreement with the incident pattern, as shown in Fig. 7.

It is of interest to note that the radial field component of the incident magnetic field—which is completely neglected in the far-field approximation—makes a noticeable contribution to the total current, as shown in Fig. 11. When $z = 5$ λ, this current contributes about 3 percent of the field strength of the scattered pattern on axis, so a small but not negligible part of the correction to the far-field approximation is the inclusion of the radial field components. However, these surface current data lead to the conclusion that the primary cause of the error resulting from the use of the far-field approximation is that the current magnitude patterns are too narrow.

The computer program developed in this report (see Appendix D) includes a subroutine FIELDS that provides the main program with values of $\mathbf{H}_i$ on the surface integration grid. Two such subroutines were written. In one subroutine the incident fields are specified by a set of spherical-wave coefficients a_n and b_n. In the other subroutine the far-field relations of Eq. (14) are assumed, and E_θ and E_ϕ are represented by exactly the same type of series as the surface data (see Eq. 12). In the case of the field data, the $m = 1$ azimuthal variation term is of special significance and is the only term present in an important class of sources (Ref. 18). For this reason, the FIELD subroutines were written for any single value m for the order of azimuthal variation. If more than one value is required, the cases may be run individually and the results superimposed. The far-field subroutine is intended for cases in which the reflecting surface is sufficiently removed from the source, or in which the available number of spherical waves (limited by storage and other considerations) is inadequate to expand the incident-field pattern. The spherical-wave subroutine should be used whenever the reflector is close to the source, and in borderline cases (which include most practical Cassegrainian antenna systems) when assured accuracy is the overriding concern.

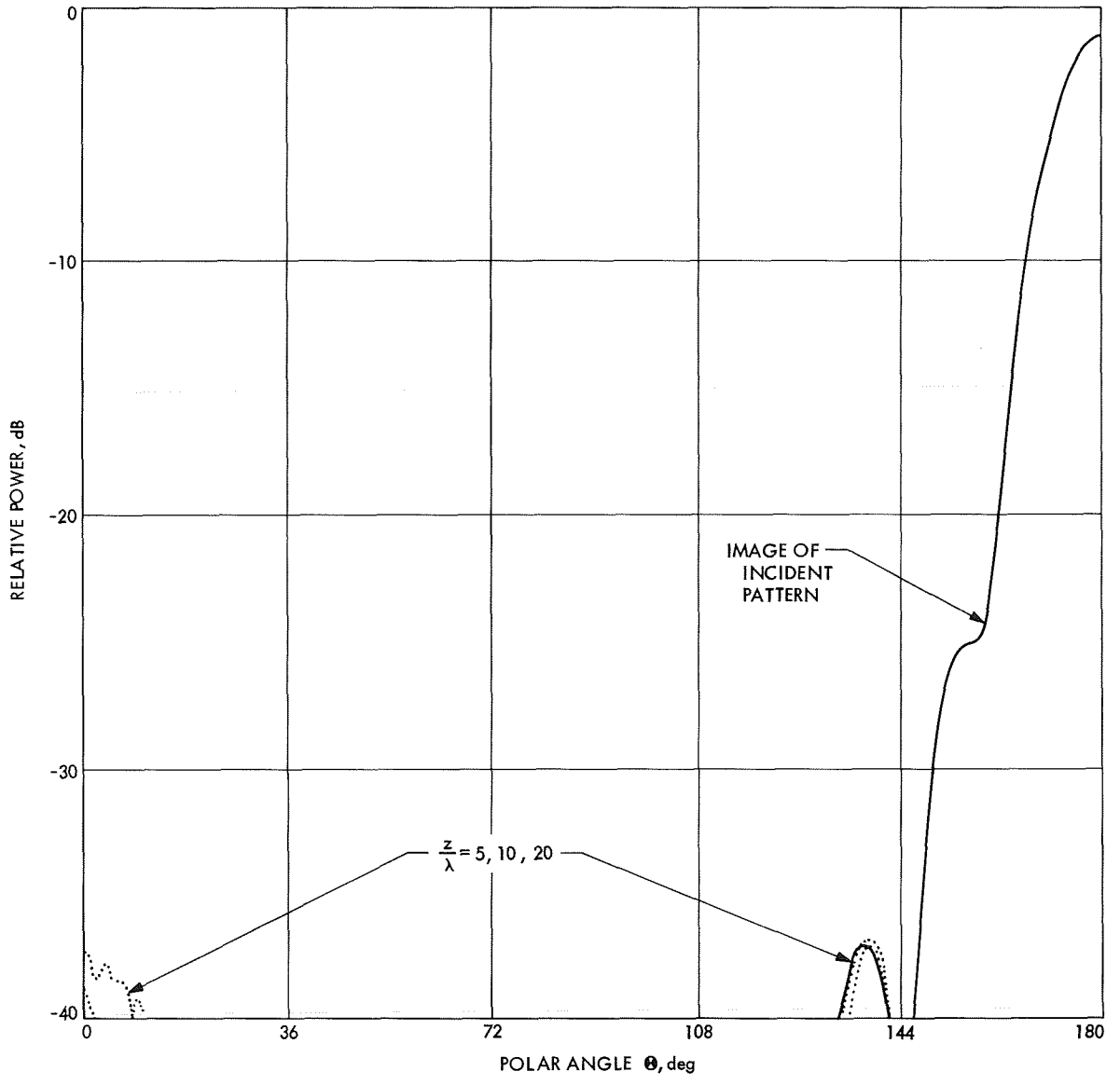


Fig. 7. Computed amplitude patterns for plane reflector cases; spherical-wave expansion of incident fields

V. Evaluation of the Integral

The numerical evaluation of the integral, Eq. (7), may justifiably be considered the most important part of the scattering problem. A primary engineering function of a scattering program is the evaluation of the effect of one or more parameters on the resulting pattern; this application is severely restricted if each case requires 6 or 8 h of computer time.

The fast Fourier transform technique has been applied to the evaluation of diffraction integrals by Pratt and Andrews (Ref. 19), but this formulation is applicable only when the argument of the exponential term is linear in each of the variables of integration. In the case considered here, the integral to be evaluated (see Eq. 7) may be written as follows:

$$\mathbf{I}(\Theta,\Phi) = -\frac{j\omega\mu}{2\pi}\int\int_S \mathbf{F}(\theta,\phi)\, e^{jk\gamma(\theta,\phi,\Theta,\Phi)}\, d\theta d\phi \qquad (16)$$

where

$$\begin{aligned}\gamma &\equiv \boldsymbol{\rho}\cdot\mathbf{i}_R - \rho \\ &= \rho\,[\sin\theta\sin\Theta\,(\cos\phi\cos\Phi + \sin\phi\sin\Phi) \\ &\quad + \cos\theta\cos\Theta - 1]\end{aligned} \qquad (17)$$

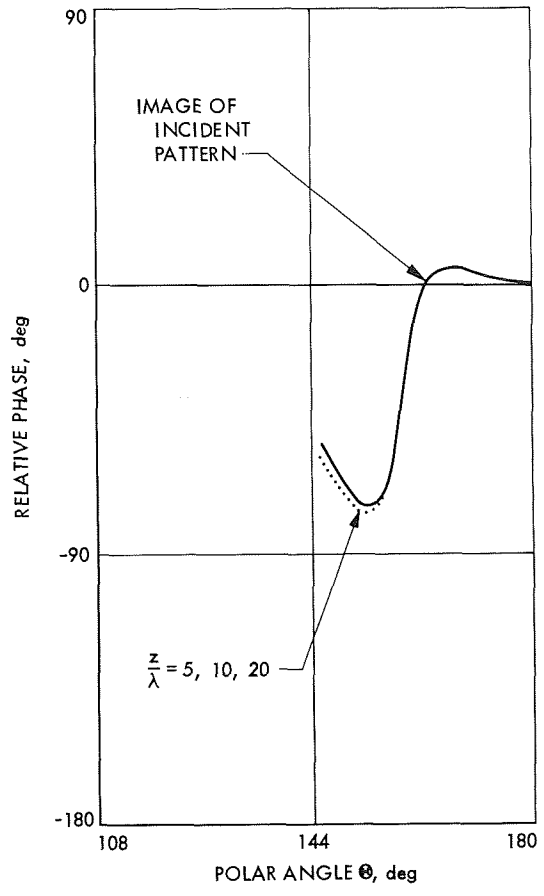


Fig. 8. Computed phase patterns for plane reflector cases; spherical-wave expansion of incident fields

Since γ is a nonlinear function of θ and ϕ, the fast Fourier transform technique is not applicable. Techniques that have been successfully applied to Eq. (16) are Gaussian quadrature, which was used by Allen (Ref. 20), and Romberg quadrature, which was recently used by Rusch and Strachman (Ref. 21). The Romberg method is particularly useful in the case of computing the main-beam and near-sidelobe regions of a high-gain pattern, and could in fact be used with the method presented below to increase efficiency in this situation.

Before describing the integration technique developed by the author, some general features involved in evaluating this integral will be discussed. Suppose $\mathbf{I}(\Theta,\Phi)$ is to be evaluated on a set of points (the output grid) given by (Θ_j, Φ_k), $1 \leq j \leq$ JMAX, $1 \leq k \leq$ KMAX. Typical values are JMAX = 91, KMAX = 2. The integral will be evaluated numerically when the integrand is specified at a set of points (the integration grid) given by (θ_m, ϕ_n), $1 \leq m \leq$ MMAX, $1 \leq n \leq$ NMAX. A modest grid size is MMAX = 50, NMAX = 181. (The effect of these grid parameters on computed pattern accuracy is discussed below and is considered in detail in Ref. 22.)

As mentioned in Section II, with the far-field assumption, $\mathbf{F}$ does not depend on Θ and Φ, so it need only be computed once on the integration grid—9050 points in this example. This operation requires little computer time, but since $\mathbf{F}$ has three Cartesian components (each of which is complex), $6 \times 9050 = 54{,}300$ values must be stored. Many computers do not have this capacity. Therefore, the program was written so that several integration grids may be specified—i.e., the reflector is divided into segments—and the resulting scattered fields from each segment are superimposed.

Unavoidably, the function $e^{jk\gamma}$ must be computed at JMAX $\times$ KMAX $\times$ MMAX $\times$ NMAX points (1,647,100 in this example). This example demonstrates the tremendous advantage of storing $\mathbf{F}$ rather than recomputing it for every output point. Also, the number of points (1,647,100) represents the number of summations involved to evaluate the integral for all of the output points. Several steps were

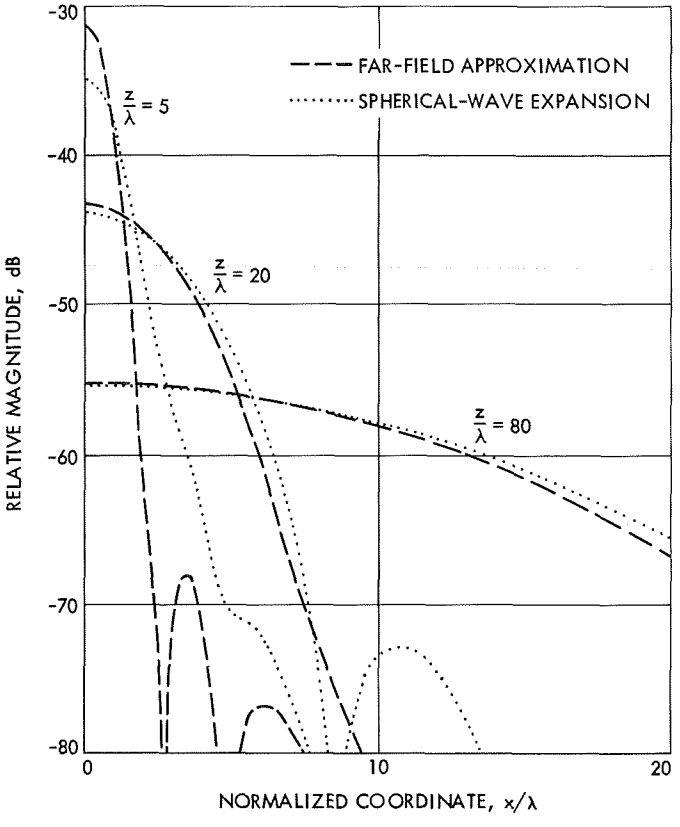


Fig. 9. Surface currents on plane reflectors; amplitude behavior

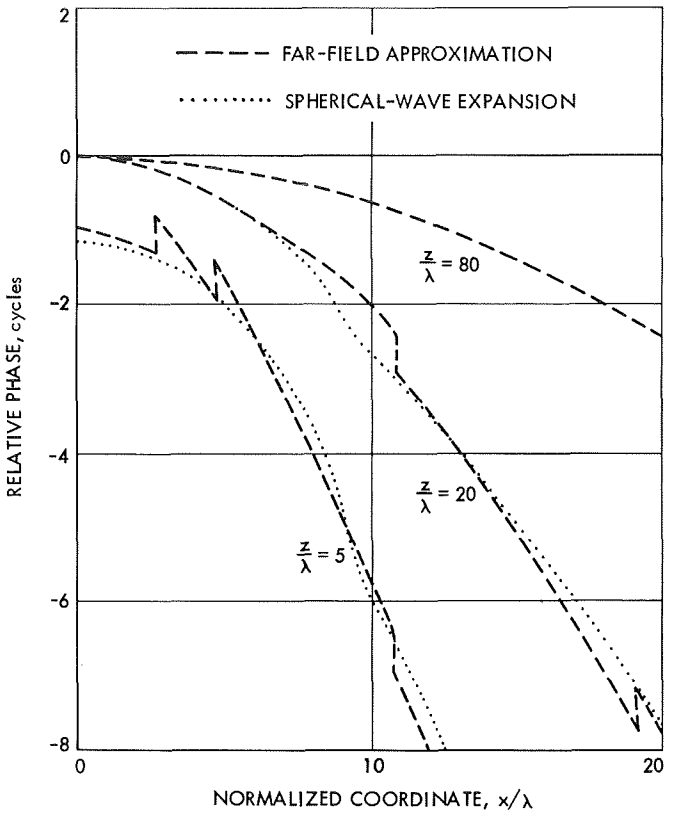


Fig. 10. Surface currents on plane reflectors; phase behavior

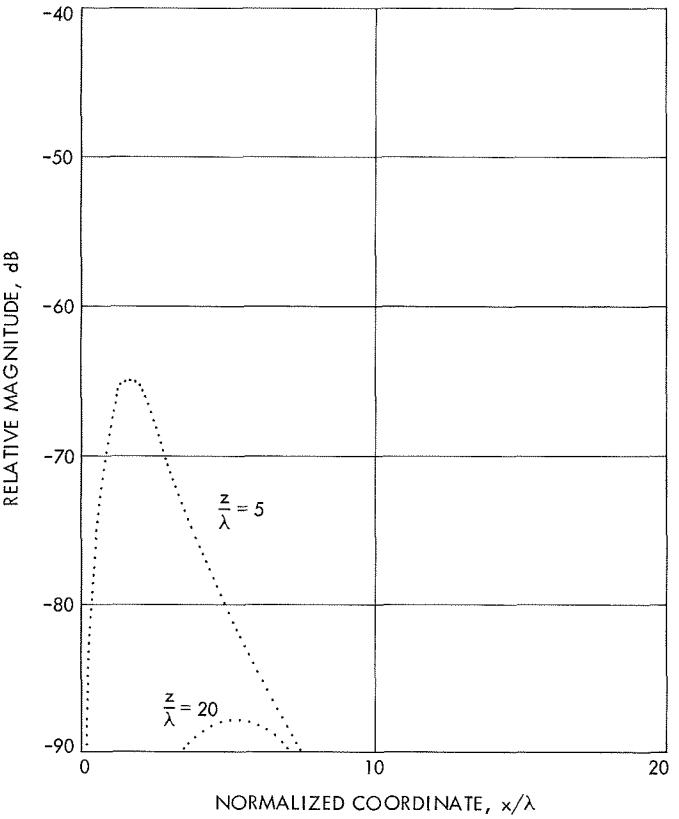


Fig. 11. Surface currents on plane reflectors; contribution of the radial component of the incident fields

taken to minimize the time used in this portion of the program:

(1) The trigonometric functions sin θ, cos θ, sin ϕ, cos ϕ, sin Θ, cos Θ, sin Φ, cos Φ are precomputed at the required grid angles and stored so that the computation of γ will involve only multiplication and addition of stored quantities, rather than the time-consuming evaluation of trigonometric functions. For this reason, Eq. (17) is not written in the form cos ϕ cos Φ + sin ϕ sin Φ = cos ($\phi - \Phi$), which would have to be computed at every permutation of ϕ and Φ values.

(2) A fast machine-language subroutine was written to compute cos $k\gamma$ + j sin $k\gamma$. This subroutine is described in Appendix B.

(3) A special numerical integration technique was developed that reduces the required values of MMAX and NMAX.

The numerical integration technique is described in Ref. 5, but for completeness the description will be partially repeated here. For a fixed output point (Θ_p, Φ_q) the integral is of the form

$$\int_{\theta_1}^{\theta_M}\int_{\phi_1}^{\phi_N} \mathbf{F}(\theta,\phi)\exp jk\gamma_{pq}(\theta,\phi)\,d\theta d\phi \tag{18}$$

where

$$\mathbf{F} = \left(\frac{\partial\rho}{\partial\phi}H_\rho + \rho\sin\theta\,H_\phi\right)\mathbf{i}_\theta + \left(-\rho\sin\theta\,H_\theta - \frac{\partial\rho}{\partial\theta}H_\rho\right)\mathbf{i}_\phi + \left(\frac{\partial\rho}{\partial\theta}\sin\theta\,H_\phi - \frac{\partial\rho}{\partial\phi}H_\theta\right)\mathbf{i}_\rho \tag{19a}$$

$$H_\rho\mathbf{i}_\rho + H_\theta\mathbf{i}_\theta + H_\phi\mathbf{i}_\phi \equiv \rho e^{jk\rho}\mathbf{H}_i \tag{19b}$$

Since the unit vectors of Eq. (19) vary over the region of integration, Cartesian components are used in the actual evaluation of the integral.

In the near field, the radial variation of $\mathbf{H}_i$ will approximate $e^{-jk\rho}/\rho$ behavior; therefore, if this dominant variation

is factored out, H_ρ, H_θ, and H_ϕ as defined in Eq. (19) will be nearly constant with respect to ρ, which makes it much easier to numerically approximate **F**. The $e^{-jk\rho}$ factor has been included in the path length function γ, and the $1/\rho$ factor is canceled by a ρ factor in the term for incremental area dS. Deviations from $e^{-jk\rho}$ phase behavior are included by allowing H_ρ, H_θ, and H_ϕ to be complex valued. (Since a radial component of $\mathbf{H}_i$ is included, this does *not* involve any assumptions or approximations).

Consider the behavior of the integrand over an incremental area of S,

$$\Delta S_{mn} = \{(\theta,\phi) : \theta_m \leq \theta \leq \theta_{m+1},\ \phi_n \leq \phi \leq \phi_{n+1}\} \tag{20}$$

Suppose that the physical dimensions of ΔS_{mn} are on the order of a wavelength, $\lambda = 2\pi/k$. Then the path length term $jk\gamma$ cannot vary by more than 2π, and since electromagnetic fields cannot generally change abruptly over distances on the order of a wavelength, **F** will not vary much. Thus, in virtually any problem, **F** and γ will be very well behaved and slowly varying over ΔS_{mn}. However, the possible 2π variation in the exponential term could cause the real and imaginary parts of the integrand to behave like a full cycle of a sinusoid. To apply a technique such as Simpson's rule to the entire integrand would require a further subdivision of ΔS. However, if the functions **F** and γ are approximated individually a simple linear form will operate satisfactorily over ΔS_{mn}. Explicitly, write

$$\mathbf{F}(\theta,\phi) \simeq \mathbf{a}_{mn} + \mathbf{b}_{mn}(\theta - \theta_m) + \mathbf{c}_{mn}(\phi - \phi_n) \tag{21a}$$

$$\gamma(\theta,\phi) \simeq \alpha_{mn} + \beta_{mn}(\theta - \theta_m) + \xi_{mn}(\phi - \phi_n) \tag{21b}$$

for $(\theta,\phi) \in \Delta S_{mn}$.

The method used for determining the coefficients is to best-fit a plane (least squares sense) to the values of the function at the corners of ΔS_{mn}. For example, in the case of the function **F** this method yields the following:

$$\mathbf{a}_{mn} = \frac{1}{4}\left[3\mathbf{F}(\theta_m,\phi_n) - \mathbf{F}(\theta_{m+1},\phi_{n+1}) + \mathbf{F}(\theta_{m+1},\phi_n) + \mathbf{F}(\theta_m,\phi_{n+1})\right] \tag{22a}$$

$$\mathbf{b}_{mn} = \frac{1}{2\Delta\theta_m}\left[\mathbf{F}(\theta_{m+1},\phi_n) - \mathbf{F}(\theta_m,\phi_n) + \mathbf{F}(\theta_{m+1},\phi_{n+1}) - \mathbf{F}(\theta_m,\phi_{n+1})\right] \tag{22b}$$

$$\mathbf{c}_{mn} = \frac{1}{2\Delta\phi_m}\left[\mathbf{F}(\theta_m,\phi_{n+1}) - \mathbf{F}(\theta_m,\phi_n) + \mathbf{F}(\theta_{m+1},\phi_{n+1}) - \mathbf{F}(\theta_{m+1},\phi_n)\right] \tag{22c}$$

where

$$\Delta\theta_m = \theta_{m+1} - \theta_m$$

$$\Delta\phi_n = \phi_{n+1} - \phi_n$$

An identical relation holds between the coefficients α_{mn}, β_{mn}, ξ_{mn} and the function γ. The integration over ΔS_{mn} may then be performed analytically, yielding the contribution:

$$\begin{aligned}\Delta\mathbf{I}_{mn} = \exp jk\alpha_{mn}\Bigg\{ & a_{mn}\left[\frac{\exp jk\beta_{mn}\Delta\theta_m - 1}{jk\beta_{mn}}\right]\left[\frac{\exp jk\xi_{mn}\Delta\phi_n - 1}{jk\xi_{mn}}\right] \\ & + \mathbf{b}_{mn}\left[\frac{\Delta\theta_m}{jk\beta_{mn}}\exp jk\beta_{mn}\Delta\theta_m - \left(\frac{\exp jk\beta_{mn}\Delta\theta_m - 1}{(jk\beta_{mn})^2}\right)\right]\left[\frac{\exp jk\xi_{mn}\Delta\phi_n - 1}{jk\xi_{mn}}\right] \\ & + \mathbf{c}_{mn}\left[\frac{\exp jk\beta_{mn}\Delta\theta_m - 1}{jk\beta_{mn}}\right]\left[\frac{\Delta\phi_n}{jk\xi_{mn}}\exp jk\xi_{mn}\Delta\phi_n - \left(\frac{\exp jk\xi_{mn}\Delta\phi_n - 1}{(jk\xi_{mn})^2}\right)\right]\Bigg\}\end{aligned} \tag{23}$$

Since it is possible for β_{mn} or ξ_{mn} to be near or equal to zero, it is necessary to develop separate equations for this case (they are easily derived from the above equation) to avoid large numerical errors.

It should be noted that many of the computations involved in evaluating each single vector component do not have to be repeated for the remaining five components. For example, in Eq. (23) only the components of the $\mathbf{a}_{mn}$, $\mathbf{b}_{mn}$ and $\mathbf{c}_{mn}$ coefficients will change. Therefore, although the six vector components present a severe storage problem, the effect on computer time is considerably less than a factor of six.

The basic idea behind this method—to isolate the oscillatory behavior of the integrand—is conceptually similar to the Eikonal technique used in deriving geometrical optics from Maxwell's equations (Ref. 23) and is also a feature of Filon's method for evaluating Fourier integrals (Ref. 24). In fact, Allen has also applied Filon's method to antenna problems and concluded that it is competitive with Gaussian quadrature (Ref. 20). The essential difference of the method presented here is that the "frequency" terms β_{mn} and ξ_{mn} are re-estimated for each incremental area of integration, rather than assumed as known constants.

If radiation integrals are considered as analytic forms of a Huygens type of principle in which infinitesimal electric or magnetic dipoles radiate a simple pattern and are summed as in an array, then another way of interpreting this technique is to describe it as replacing infinitesimal elements with elements about a wavelength square that radiate a more complicated pattern as given by Eq. (23).

Although this technique appears obvious and has an intuitive appeal from an engineering standpoint, it is somewhat unusual mathematically. The integral is a linear operator, and virtually all quadrature formulas are also linear, except this one. The easiest way to show this is to note that the functions 1 and $e^{jk\theta}$ are both of the form for which the technique will be exact. However, their sum

$$1 + e^{jk\theta} = 2\cos\frac{k\theta}{2}\, e^{jk\theta/2} \tag{24}$$

is not of the form $(a + b\theta)\, e^{jk(\alpha+\beta\theta)}$, and the sum of the (numerical) integrals is not equal to the (numerical) integral of the sum. However, if $\mathbf{F}$ and γ are well behaved, the technique does in fact converge to the integral operator (and therefore becomes linear) in the limit of zero step size (Ref. 5).

Convergence tests for the case of scattering from a hyperboloid show that with this technique, incremental areas 2/3 of a square wavelength in size result in errors more than 40 dB below the pattern maxima (Ref. 22). Similar tests of a program that uses Simpson's rule to evaluate a one-dimensional integral show that the areas must be at most 0.04 square wavelengths for the two-dimensional case, and possibly smaller (Ref. 22). Therefore, the number of integration points may be reduced by at least a factor of 16 relative to Simpson's rule. The plane reflector data presented in a previous section were computed with the step sizes given in Ref. 22. These data are an absolute test of numerical accuracy and are further confirmation of the convergence tests.

Although the complicated form of Eq. (23) seems to indicate that this technique would require an order of magnitude more time per data point than Simpson's rule, the fact is that in this type of problem the evaluation of $e^{jx} = \cos x + j \sin x$ tends to dominate machine time. For example, an IBM 7094 will multiply or divide two numbers in about 10 μs, or add two numbers in about 15 μs (Ref. 25). However, computation of the sine and cosine to obtain the real and imaginary parts of e^{jx} requires 571 μs with the library subroutine, or 281 μs with the fast subroutine described in Appendix B (see Table B-1).

It is estimated that time per data point for this technique is increased by a factor of 2 to 4, relative to Simpson's rule, which means that the net reduction in total running time is a factor of 4 to 8. In practice, this technique can result in a reduction from 8 h to 1 or 2 h of computer time, so it is of substantial practical importance. For example, Slobin (Ref. 16) has reported that a program using Simpson's rule integration requires 1.3 min of IBM 360/75 computer time per output point to compute the scattered pattern of a 56-wavelength-diam tilted subreflector. Thus, 3.90 h would be required for 180 output points. These data have been obtained in 1.01 h of IBM 7094 Mod I computer time with the program described here. (The 1.01 h includes 9.8 min to evaluate the spherical-wave expansion of the incident fields at each of the integration grid points.) The 360/75 is faster than the 7094 Mod I by a factor of between 3 and 5; therefore the program described here is 11 to 19 times faster than this particular program that uses Simpson's rule. (It is interesting to note that the new technique works better on an old computer than the old technique on a new computer.)

It should be noted that in the case of rotational symmetry, when one integration may be performed analytically, these same data can be obtained by evaluating a one-dimensional integral (with a Bessel function in the

integrand) in about 6 min (Ref. 22). Therefore, it is grossly inefficient to evaluate a double integral in symmetrical cases, and the program described here should only be applied to problems involving asymmetrical configurations.

These convergence data are representative of the "low-gain" case in which the contributions from currents on the surface do not add in phase at any output point. A special case of considerable practical interest is the computation of the main lobe and first few sidelobes of the scattered pattern of a high-gain antenna such as a paraboloidal reflector. In this case, contributions are adding nearly in phase, i.e., the phase of the integrand is *very* slowly varying. Tests recently completed indicate that, in this case, integration points may be spaced at intervals of 20 wavelengths or more without seriously affecting the accuracy of the results. (Actually, in many practical cases closer spacing will be required to represent sufficient detail of the surface and incident-field data.) The results of these recent tests are similar to results reported by Allen for Filon's method and Gaussian quadrature for this case (Ref. 20), so the integration technique used here is nearly equivalent to the techniques in this situation. As mentioned previously, any of these methods may be improved by a Romberg type of modification in this case. The large spacings allowable between grid points means that the patterns of reflectors over 1000 wavelengths in diameter may be computed. However, beyond the first few sidelobes the situation begins to approach the "low-gain" case described earlier.

VI. Comparison With Experimental Data and Other Results

An improved theory is important if it confirms the results of an approximate theory, since the approximation may then be used with confidence; it is also important if an improved theory corrects the results of an approximate theory. The application of spherical-wave theory developed in this report is considered valuable on both counts. A case in which the far-field approximation yields an incorrect result has already been presented; a case in which this approximation is adequate, except for the most precise calculations, is discussed in this section of the report.

A basic criterion in selecting this sample case was that it represent a real problem. The standard NASA/JPL Deep Space Network antenna is an 85-ft-diam paraboloid, with a Cassegrainian feed system, operated at S-band. This configuration is also common for communications satellite terminals, so this sample case is clearly one that is of practical interest.

The object of the analysis was to compute the radiation pattern of the overall antenna, with particular emphasis on the pattern maximum that determines the antenna gain. The antenna is illustrated in Fig. 12. The design of this particular antenna is well documented (Ref. 26), and JPL provided experimental patterns of the fields scattered by the subreflector used with this configuration.

There is reason to question the application of the far-field approximation to the analysis of this antenna, because the subreflector is at 1.65 D^2/λ of the primary feed, and the main reflector is at 0.148 D^2/λ of the subreflector. Because the results of a far-field analysis generally agreed with experimental data, it was certain that the approximation was reasonably good, but the amount of error introduced was uncertain until the completion of the analysis presented in this report.

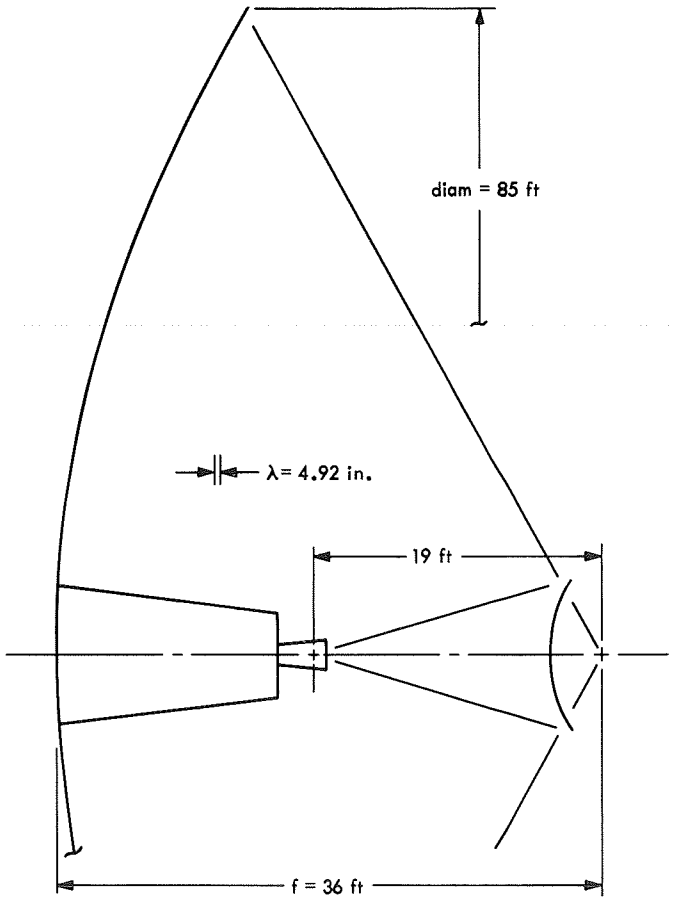


Fig. 12. Configuration of the 85-ft-diam antenna

The primary feed for this antenna was the same dual mode conical horn (Ref. 27) used for the plane reflector cases. The spherical-wave coefficients for this feed were obtained from an experimental pattern using the technique discussed in Appendix A; the incident-field patterns shown in Figs. 7 and 8 represent an evaluation of the spherical-wave expansion at $\rho = \infty$. Two parallel analyses were performed on the antenna. One analysis was made with the spherical-wave expansion FIELDS subroutine and the other with the far-field approximation FIELDS subroutine. The analyses were identical in all other respects.

The two computed scattered patterns of the subreflector (which consists of a vertex plate, a hyperboloidal section, and a conical flange) were virtually identical. This is not surprising, because the subreflector is near the $2D^2/\lambda$ distance from the primary feed.

The experimental subreflector pattern was measured at a range that was nearly equal to the distance to the main reflector. In Fig. 13, these experimental data show good agreement with the pattern computed with the far-field approximation. The pattern computed using a spherical-wave expansion was in turn expanded in a series of spherical waves. In Fig. 14, the spherical-wave expansion, evaluated at the same radius at which the experimental data were measured, is compared to the experimental pattern. The agreement is even better than that in Fig. 13. The computed data closely follows the experimental pattern, demonstrating that the agreement between theory and experiment is outstanding.

The next step in the analysis was to use the subreflector pattern to illuminate the main paraboloidal reflector, again by the use of a spherical-wave expansion and the far-field approximation in the respective cases. The two computed main reflector patterns are compared in Fig. 15. Although the most noticeable difference is in the sidelobes, this difference will generally be of little importance. Of greatest significance is that the peak of the main beam (i.e., the computed gain) differs by 0.127 dB in the two cases.

With the scale used in Fig. 15, a 0.127-dB difference is difficult to detect, and in many applications of computed antenna data it will be of no importance whatsoever. Therefore, for these applications (and for antennas of this approximate design) it has been shown that the faster and simpler far-field analysis is inadequate. For example, the far-field analysis will be used for the feed displacement calculations to be presented shortly. Use of the far-field analysis is of particular importance in the case of development work involving a large amount of computer time.

However, there are cases in which a 0.127-dB difference is significant. Computed data have reached a level of sophistication where they are frequently weighted about equally with experimental data in calibrating large ground antennas (Ref. 28). An 85-ft antenna costs about one million dollars (Ref. 29), and it can be argued (Ref. 30) that an uncertainty of 0.127 dB in the gain of such an antenna is worth $40,000. In fact, the costs of the time and effort spent to calibrate an antenna with this precision are even greater than $40,000. Since the IBM 7094 computer time required for the two cases shown in Fig. 15 (including the spherical-wave expansion of the subreflector pattern) was 23.24 and 8.60 min for the spherical-wave and far-field analysis, respectively (a difference of about $61.00 at the current price of computer time) it is probably safe to say that most engineers would choose the more accurate analysis. (Since the evaluation of the spherical-wave expansion on the integration grid is independent of the number of output points, the relative times would be more nearly equal for a larger number of output points.)

In all of the cases evaluated so far, rotational symmetry has been assumed. As a last example, the problem of an offset feed system will be considered to illustrate some of the capabilities of the program for asymmetric geometries.

The case considered is the offset feed geometry shown in Fig. 16. To provide a basis of comparison, the displacements along the dashed curve shown in the figure are the same as a case previously analyzed by Ruze (Ref. 31). However, the true subreflector pattern has been retained, thereby eliminating the need to assume a simple form for the illumination of the main reflector. In addition to the displacement translations, the feed system was rotated so that the edge angle of the main reflector remained nearly constant, to minimize spillover. The offset angle β is measured in the conventional manner. The computed patterns for a zero offset and two nonzero offsets are shown in Fig. 17. The analysis given by Ruze, which is based on scalar theory, includes the case of an illumination function of the form $f(x) = 0.3 + 0.7(1 - x^2)$, that approximates the reflector illumination in the case considered here.

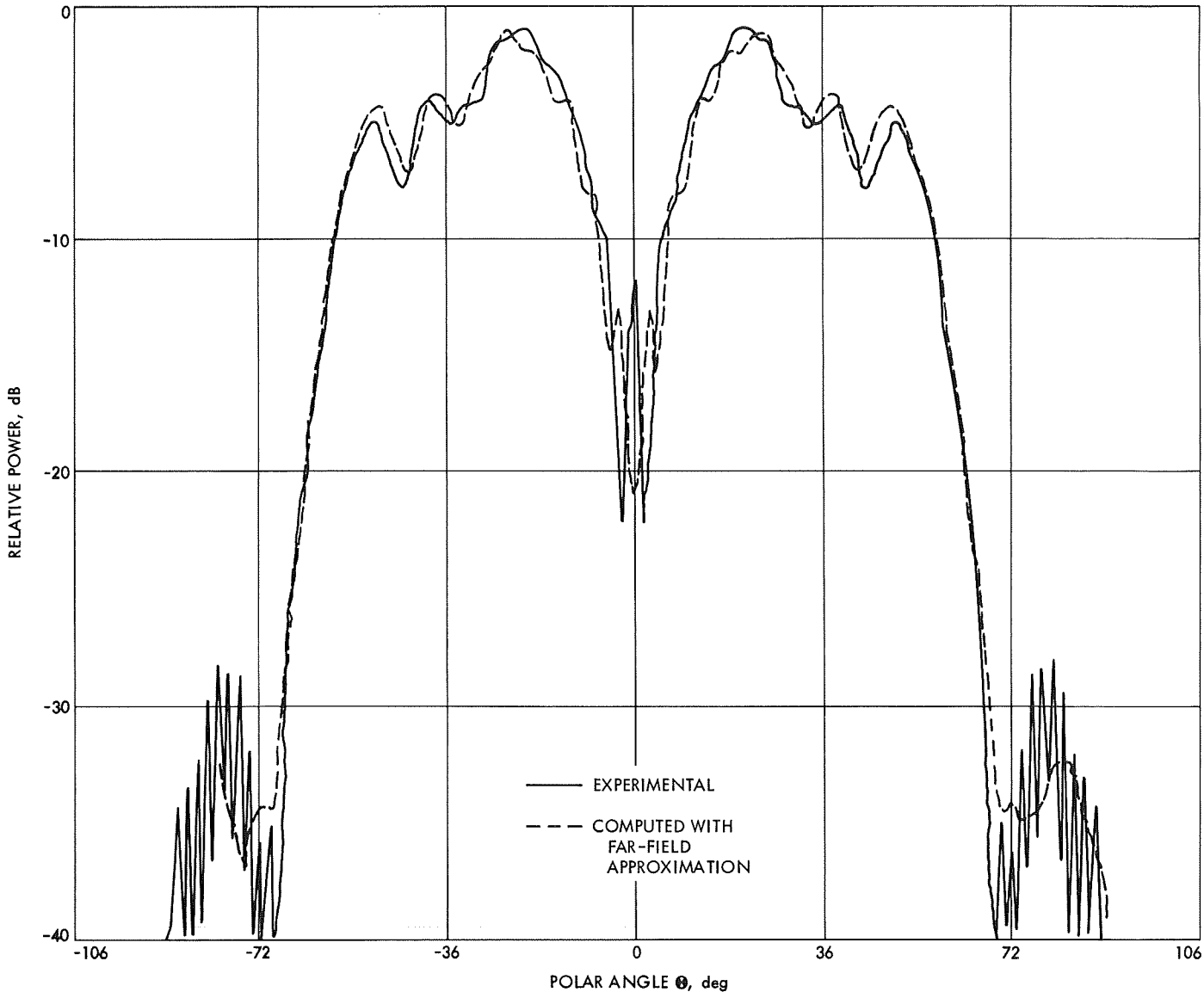


Fig. 13. Comparison of experimental and computed subreflector patterns; far-field approximation case

For this geometry and illumination, Ruze predicted a beam deviation factor of 0.84. The beam deviation factors of the results given in Fig. 17 are 0.835 and 0.840, for the 1- and 2-deg cases, respectively. The coma lobes agree within 1 dB with the values predicted by Ruze and the half and tenth-power beamwidths agree within 0.04 deg. The gain loss agrees (as well as it is possible to read the graphs given by Ruze) within approximately 0.1 dB.

This agreement is so complete that it might seem irrelevant to bother with the more rigorous analysis. However, in addition to the confidence factor mentioned earlier, the approximate analysis has several restrictions: (1) it assumes a simple analytic illumination function, (2) the displacements *must* be along the specific curve shown in Fig. 16, and (3) the displacements are assumed to be small compared to the focal length. In the case considered in this section, these restrictions were met by design to obtain a comparison with independent data, but the program is capable of a case with (1) a very complicated illumination pattern (for example, a pattern with a central null produced by a vertex plate, including the phase pattern perturbations typical in this situation), (2) the displacements in any direction and, (3) a tilt

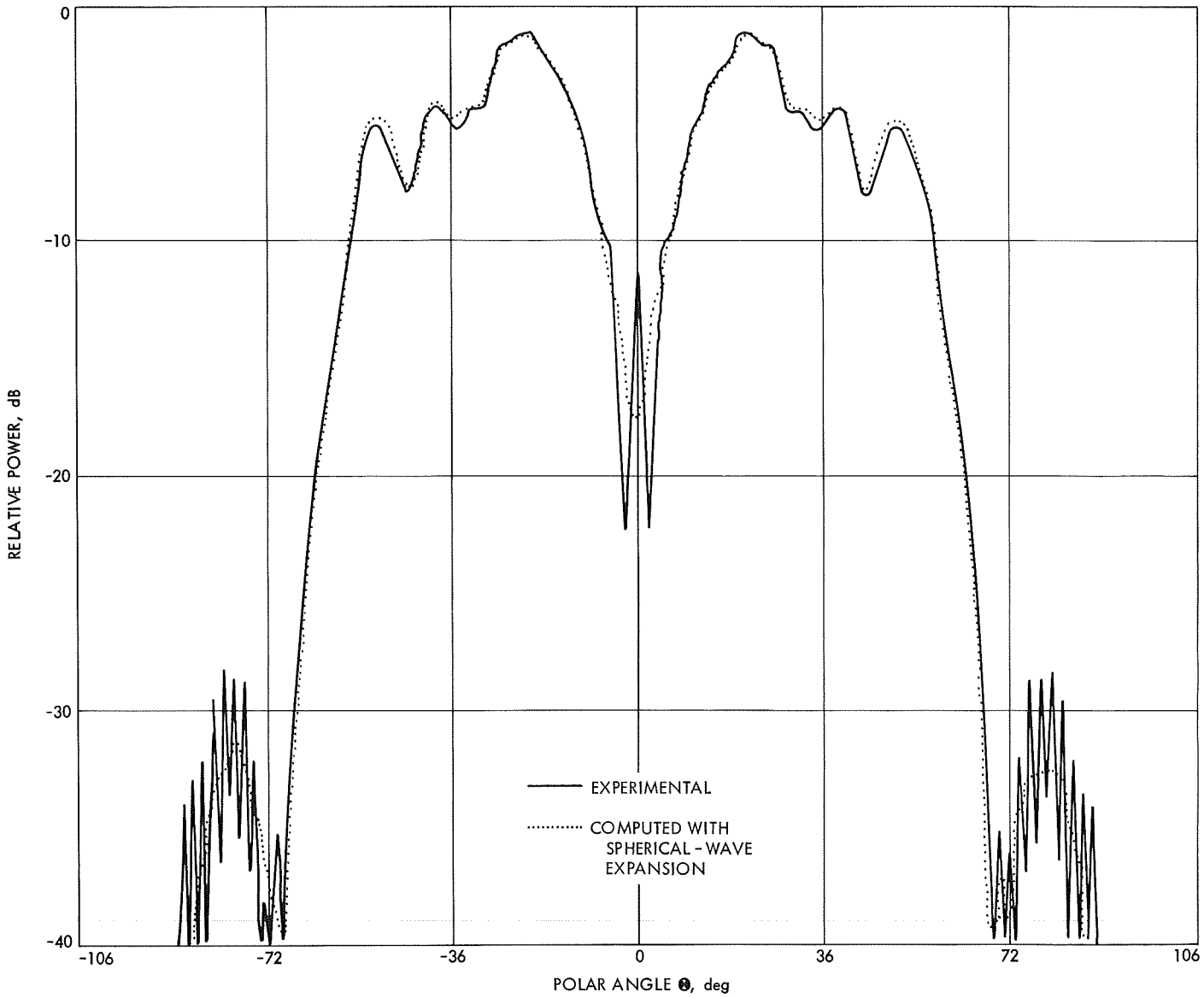


Fig. 14. Comparison of experimental and computed subreflector patterns; spherical-wave expansion case

angle that is restricted to approximately 5 deg in the present program (the wavelength determines the allowable peak error that in turn determines the tilt angle as shown in Fig. 2), but that could be made arbitrarily large if more Fourier components were included, or if a special surface subroutine were written for the particular case of a tilted paraboloid.

VII. Summary

The results shown in the previous section clearly demonstrate the usefulness of the physical-optics technique. The agreement between theoretical and experimental data is excellent, and in the case of the 85-ft antenna, the application of a technique such as the integral-equation method to obtain the surface currents is virtually hopeless because of the large size of the reflector.

It was demonstrated in Section IV that the far-field approximation for the incident fields can lead to poor results, and that the spherical-wave representation corrects the results. The far-field expansion is still useful in many practical cases, but if precise results are required, and if the reflectors are in or close to the near-field region, the spherical-wave representation should be used.

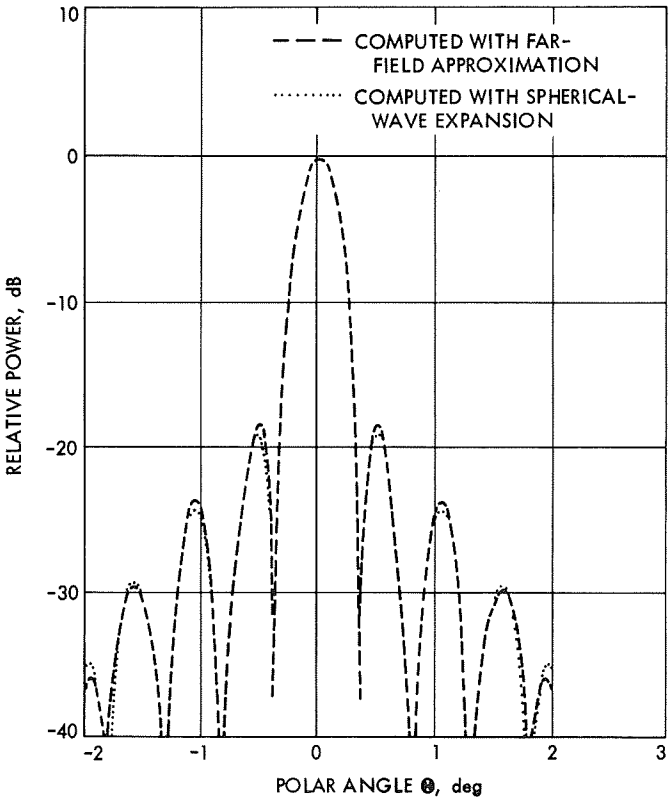


Fig. 15. Computed patterns of 85-ft-diam main reflector; comparison of far-field and spherical-wave expansion cases

A Fourier type of representation of the scattering surface is efficient in the case of a slightly tilted figure of revolution, and in many cases three components are sufficient to accurately specify the surface.

In the case of a low-gain asymmetrical reflector the program developed here is an order of magnitude faster than an earlier program; in the high-gain case the integration technique loses much of its advantage over prior methods, but the other timesaving techniques used still result in efficient operation. For symmetrical geometries, one integration should be performed analytically.

VIII. Possible Improvements and Future Applications of Computer Program

The computer program could be improved in several ways. For example, a Romberg type of procedure could be incorporated into the integration method. Also, in the existing program, when the edge of the reflector is not coincident with a coordinate boundary, an approximate boundary must be constructed out of trapezoidal-like segments. An interesting extension of the integration method developed here would be to compute the scattered pattern of a triangular-shaped segment to be used in constructing edges that are at least continuous. Another possible improvement is to modify the program to accept an arbitrary number of Fourier components in the surface representation that would eliminate the small restriction on the reflector tilt angles.

The main area of future work lies in applying the scattering program. Parameter studies of near-field Cassegrainian antennas using actual experimental feed patterns is an obvious example. In all of the cases considered above, the spherical-wave coefficients of an incident-field pattern were obtained with the program described in Appendix D, but the scattering program will accept coefficients regardless of how they are obtained. For example, if the currents excited on a feed device, such as log-periodic wire antenna, are known (perhaps from the integral equation method), the spherical-wave coefficients may be obtained directly from the currents.

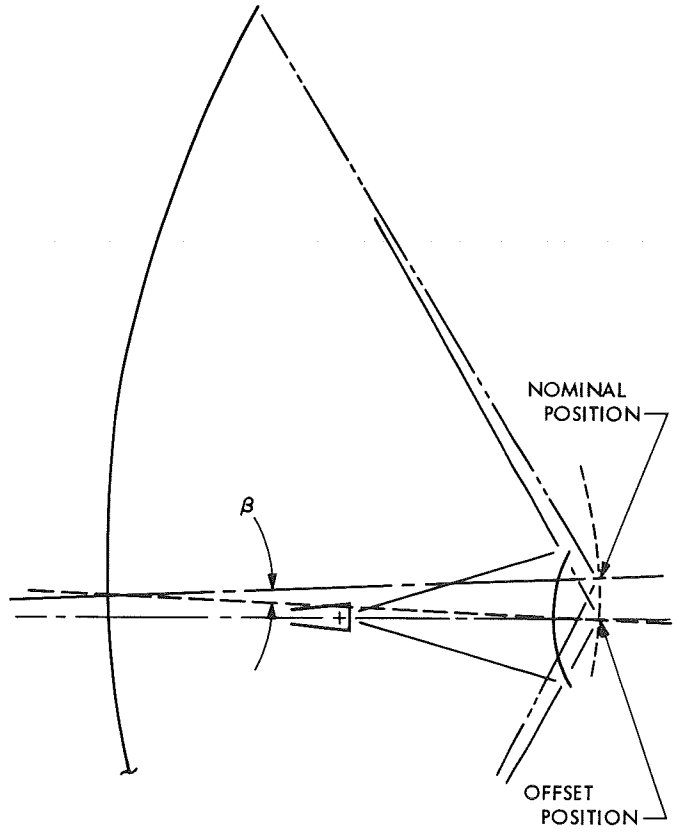


Fig. 16. Geometry of offset-feed cases

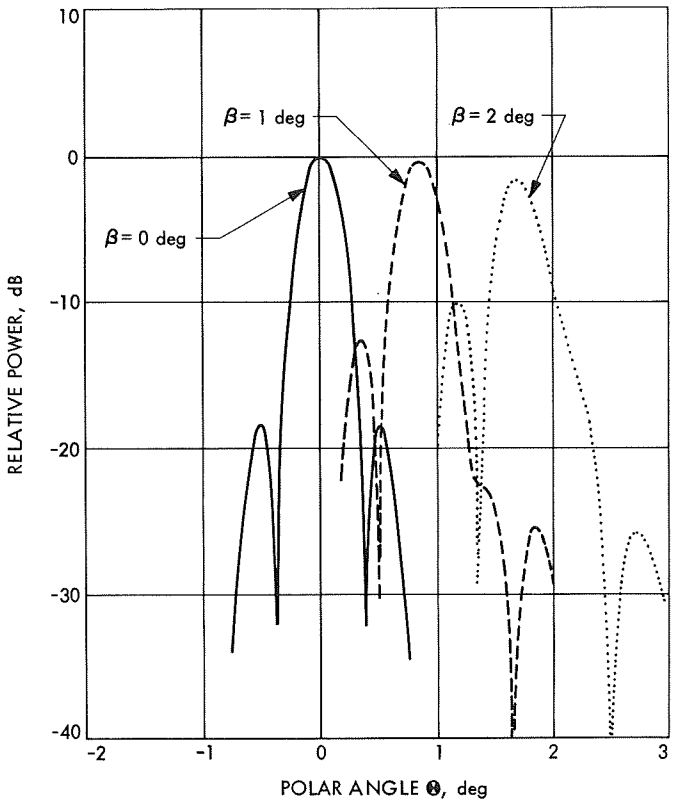


Fig. 17. Computed patterns of 85-ft-diam main reflector with offset feed

Also spherical-wave coefficients have been obtained analytically for cases such as a plane wave and an arbitrarily located and oriented dipole (Ref. 32). This dipole case is precisely the expansion required to compute the fields in the focal region of a reflector by the use of the same reciprocity technique recently used by Rusch (Ref. 21). This is a very interesting area for future work, since focal-region fields studied have involved assumptions and restrictions, whereas this method is, in principle, exact. Also, this method is applicable to nonparaboloidal antennas, such as shaped Cassegrainian antennas (Ref. 33), that have never been analyzed to obtain the focal-region fields. The reciprocal part of this problem is the computation of the pattern of reflectors with defocused feeds, and this is also a very interesting area for application of the scattering program, as discussed in the previous section.

The spherical-wave expansion program also has future applications independent of the scattering program. For example, a study is currently being pursued to apply spherical-wave expansions to the determination of correction factors for near-field gain measurements (Ref. 34). The usefulness of mode expansions in the solution of boundary-value problems has not been exhausted, and cases may be found in which the method is superior to the currently popular integral-equation method.

Appendix A

Spherical-Wave Expansions of Electromagnetic Fields

In a source-free, homogeneous, isotropic medium, spherical-wave solutions of Maxwell's equations are given by Stratton (Ref. 35):

$$\mathbf{m}^{e}_{\substack{o\\mn}} = \mp z_n(k\rho)\frac{mP^m_n(\cos\theta)}{\sin\theta}\begin{matrix}\sin\\\cos\end{matrix} m\phi\,\mathbf{i}_\theta - z_n(k\rho)\frac{\partial}{\partial\theta}P^m_n(\cos\theta)\begin{matrix}\cos\\\sin\end{matrix} m\phi\,\mathbf{i}_\phi \tag{A-1a}$$

$$\mathbf{n}^{e}_{\substack{o\\mn}} = n(n+1)\frac{z_n(k\rho)}{k\rho}P^m_n(\cos\theta)\begin{matrix}\sin\\\cos\end{matrix} m\phi\,\mathbf{i}_\rho + \frac{1}{k\rho}\frac{\partial}{\partial\rho}[\rho z_n(k\rho)]\frac{\partial}{\partial\theta}P^m_n(\cos\theta)\begin{matrix}\sin\\\cos\end{matrix} m\phi\,\mathbf{i}_\theta \pm \frac{1}{k\rho}\frac{\partial}{\partial\rho}[\rho z_n(k\rho)]\frac{mP^m_n(\cos\theta)}{\sin\theta}\begin{matrix}\cos\\\sin\end{matrix} m\phi\,\hat{\mathbf{i}}_\phi \tag{A-1b}$$

where

$e^{j\omega t}$ = time dependence (implicitly assumed)

(ρ,θ,ϕ) = spherical coordinates (see Fig. 1)

$k = \omega(\epsilon\mu)^{1/2} = 2\pi/\lambda$ = propagation constant

$z_n(k\rho)$ = any solution of the spherical Bessel equation

$P^m_n(\cos\theta)$ = associated Legendre function

In the following derivation, an electromagnetic field is understood to be a pair of vector valued functions **E** and **H** (defined everywhere in a source-free, linear, homogeneous, isotropic region V) that satisfies

$\nabla \times \mathbf{E}$ and $\nabla \times \mathbf{H}$ exist everywhere in V

and

$$\left.\begin{aligned}\nabla\times\mathbf{E} &= -j\omega\mu\mathbf{H}\\ \nabla\times\mathbf{H} &= j\omega\epsilon\mathbf{E}\end{aligned}\right\} \tag{A-2a}$$

where ϵ and μ are physical constants, and ω is a nonzero positive constant, and where

$$\left.\begin{aligned}&\int_V \epsilon\,|\mathbf{E}|^2\,dv\\ &\int_V \mu\,|\mathbf{H}|^2\,dv\end{aligned}\right\} \tag{A-2b}$$

are defined and finite.

Since $\nabla\cdot\nabla\times\mathbf{A} = 0$ for any vector **A**, Eq. (A-2a) is equivalent to the assumption that Maxwell's equations are satisfied; by Poynting's theorem, Eq. (A-2b) is equivalent to the requirement that sources radiate finite power (assuming finite energy in any resonant fields).

The parallels between electromagnetic theory and the theory of complex variables are intriguing, and it is interesting to note that by Eq. (A-2a), the existence of the first derivative ($\nabla\times\mathbf{E}$ and $\nabla\times\mathbf{H}$) assures the existence of derivatives of all orders ($\nabla\times\nabla\times\cdots\times\nabla\times\mathbf{E}$, etc.). Other striking analogies are between the field integrals (which express the fields everywhere in a volume in terms of the values on the surface enclosing the volume) and Cauchy's theorem, and between spherical-wave expansions and Laurent series.

The objective of this section is as follows: given an electromagnetic field in a region V that consists of all space outside of a sphere of radius ρ_0 (i.e., all sources are enclosed in this sphere), determine coefficients $a^{e}_{\substack{o\\mn}}$ and $b^{e}_{\substack{o\\mn}}$ so that everywhere in V

$$\mathbf{E}(\rho,\theta,\phi) = -\sum_m\sum_n a^{e}_{\substack{o\\mn}}\mathbf{m}^{e}_{\substack{o\\mn}} + b^{e}_{\substack{o\\mn}}\mathbf{n}^{e}_{\substack{o\\mn}} \tag{A-3a}$$

$$\mathbf{H}(\rho,\theta,\phi) = \frac{k}{j\omega\mu}\sum_m\sum_n a^{e}_{\substack{o\\mn}}\mathbf{n}^{e}_{\substack{o\\mn}} + b^{e}_{\substack{o\\mn}}\mathbf{m}^{e}_{\substack{o\\mn}} \tag{A-3b}$$

Jones (Ref. 32) has shown that any electromagnetic field can be written in this form. Furthermore, since sources were assumed to be of a finite extent, only the solutions involving $h^{(2)}_n(k\rho)$ (the spherical Hankel function of the second kind that satisfies the radiation condition and corresponds to an outward traveling wave, as discussed in Refs. 36 and 37) need be included.

Bouwkamp and Casimir (Ref. 38) obtained expressions for the coefficients in terms of the source currents; Kennaugh (Ref. 36) gives expressions in terms of the tangential **E** and **H** fields on a surface enclosing the sources; Jones (Ref. 32) includes expressions in terms of all components of either **E** or **H**. Since the data involved here are the tangential components of **E** on a sphere of radius $\rho_1 > \rho_0$, a derivation for this case will be given. (Note that the case Jones considers is equivalent for the determination of TE wave coefficients but not for the determination of TM wave coefficients.)

As a first step, let a^{e}_{omn} and b^{e}_{omn} be any set of coefficients so that Eq. (A-3) is satisfied. (As mentioned above, Jones has shown at least one such set must exist.)

Define

$$a'^{e}_{omn} \equiv a^{e}_{omn}\, h^{(2)}_n (k\rho_1) \tag{A-4a}$$

$$b'^{e}_{omn} \equiv b^{e}_{omn} \frac{1}{k\rho_1} \frac{\partial}{\partial \rho} [\rho h^{(2)}_n (k\rho)]_{\rho=\rho_1} \tag{A-4b}$$

Then, since Eq. (A-3) is true everywhere in V, in particular, there must be equality of the tangential components at $\rho = \rho_1$:

$$E_\theta(\theta,\phi) = \sum_m \sum_n \mp a'^{e}_{omn} \frac{mP^m_n(\cos\theta)}{\sin\theta} \begin{matrix}\sin\\ \cos\end{matrix} m\phi + b'^{e}_{omn} \frac{\partial}{\partial\theta} P^m_n(\cos\theta) \begin{matrix}\sin\\ \cos\end{matrix} m\phi \tag{A-5a}$$

$$E_\phi(\theta,\phi) = \sum_m \sum_n - a'^{e}_{omn} \frac{\partial}{\partial\theta} P^m_n(\cos\theta) \begin{matrix}\cos\\ \sin\end{matrix} m\phi \pm b'^{e}_{omn} \frac{mP^m_n(\cos\theta)}{\sin\theta} \begin{matrix}\cos\\ \sin\end{matrix} m\phi \tag{A-5b}$$

where

$$E_\theta(\theta,\phi) \equiv \mathbf{E}(\rho_1,\theta,\phi) \cdot \mathbf{i}_\theta$$

$$E_\phi(\theta,\phi) \equiv \mathbf{E}(\rho_1,\theta,\phi) \cdot \mathbf{i}_\phi$$

The azimuthal components are separated out by the use of an ordinary Fourier expansion. (These equations are true for $m > 0$. For $m = 0$, an additional factor of 2 should be included but has been omitted for clarity.)

$$A^{e}_{mo}(\theta) \equiv \int_0^{2\pi} E_\theta(\theta,\phi) \begin{matrix}\sin\\ \cos\end{matrix} m\phi\, d\phi = -\pi \sum_n \mp a'^{e}_{omn} \frac{mP^m_n(\cos\theta)}{\sin\theta} + b'^{e}_{omn} \frac{\partial}{\partial\theta} P^m_n(\cos\theta) \tag{A-6a}$$

$$B^{e}_{mo}(\theta) \equiv \int_0^{2\pi} E_\phi(\theta,\phi) \begin{matrix}\cos\\ \sin\end{matrix} m\phi\, d\phi = -\pi \sum_n - a'^{e}_{omn} \frac{\partial}{\partial\theta} P^m_n(\cos\theta) \pm b'^{e}_{omn} \frac{mP^m_n(\cos\theta)}{\sin\theta} \tag{A-6b}$$

For the next step, the following relationship is needed:

$$\int_0^\pi \left[\frac{mP^m_\ell(\cos\theta)}{\sin\theta} \pm \frac{\partial}{\partial\theta} P^m_\ell(\cos\theta)\right]\left[\frac{mP^m_n(\cos\theta)}{\sin\theta} \pm \frac{\partial}{\partial\theta} P^m_n(\cos\theta)\right] \sin\theta\, d\theta =$$

$$\int_0^\pi \left[\frac{m^2 P^m_\ell P^m_n}{\sin^2\theta} + \frac{\partial}{\partial\theta} P^m_\ell \frac{\partial}{\partial\theta} P^m_n\right] \sin\theta\, d\theta \pm \int_0^\pi \left[\frac{\partial}{\partial\theta} P^m_\ell \frac{mP^m_n}{\sin\theta} + \frac{mP^m_\ell}{\sin\theta} \frac{\partial}{\partial\theta} P^m_n\right] \sin\theta\, d\theta =$$

$$\begin{cases} 0 & \text{for } \ell \neq n \\ \dfrac{2}{2n+1} \dfrac{(n+m)!}{(n-m)!} n(n+1) & \text{for } \ell = n \end{cases} \tag{A-7}$$

The equation above may be obtained by the performance of one integration by parts

$$\int_0^\pi \left[\frac{\partial}{\partial\theta} P^m_\ell \frac{mP^m_n}{\sin\theta} + \frac{mP^m_\ell}{\sin\theta} \frac{\partial}{\partial\theta} P^m_n\right] \sin\theta\, d\theta = \int_0^\pi \left[\frac{\partial}{\partial\theta} P^m_\ell \frac{mP^m_n}{\sin\theta}\right] \sin\theta\, d\theta + [mP^m_\ell P^m_n]_0^\pi - \int_0^\pi \left[\frac{\partial}{\partial\theta} P^m_\ell \frac{mP^m_n}{\sin\theta}\right] \sin\theta\, d\theta = 0 \tag{A-8}$$

and application of a result given by Stratton (Ref. 35) (Section 7.13, Chapter VII).

Then it is easily verified that

$$\int_0^\pi \left[\pm \frac{mP_\ell^m(\cos\theta)}{\sin\theta} A_{m_o^e}(\theta) + \frac{\partial}{\partial\theta} P_\ell^m(\cos\theta) B_{m_o^e}(\theta) \right] \sin\theta\, d\theta = \pi a'_{o^e m\ell} \frac{2}{2\ell+1} \frac{(\ell+m)!}{(\ell-m)!} \ell(\ell+1) \tag{A-9a}$$

$$\int_0^\pi \left[-\frac{\partial}{\partial\theta} P_\ell^m(\cos\theta) A_{m_o^e}(\theta) \mp \frac{mP_\ell^m(\cos\theta)}{\sin\theta} B_{m_o^e}(\theta) \right] \sin\theta\, d\theta = \pi b'_{o^e m\ell} \frac{2}{2\ell+1} \frac{(\ell+m)!}{(\ell-m)!} \ell(\ell+1) \tag{A-9b}$$

Equations (A-4) through (A-9) may be rewritten in the expected form:

$$a_{o^e m\ell} = \frac{1}{[z_n(k\rho_1)]^2} \frac{2n+1}{\pi 2n(n+1)} \frac{(n-m)!}{(\ell+m)!} \times \int_0^{2\pi}\int_0^\pi -\mathbf{m}_{o^e mn} \cdot \mathbf{E}_i(\rho_1,\theta,\phi)_{\text{tangential}} \sin\theta\, d\theta\, d\phi \tag{A-10a}$$

$$b_{o^e m\ell} = \frac{1}{\left\{\frac{1}{k\rho_1}\frac{\partial}{\partial\rho}[\rho_1 z_n(k\rho_1)]\right\}^2} \frac{2n+1}{\pi 2n(n+1)} \frac{(n-m)!}{(n+m)!} \times \int_0^{2\pi}\int_0^\pi -\mathbf{n}_{o^e mn} \cdot \mathbf{E}_i(\rho_1,\theta,\phi)_{\text{tangential}} \sin\theta\, d\theta\, d\phi \tag{A-10b}$$

Equation (A-10) not only gives an explicit form for the desired coefficients, but, by the well known properties of trigonometric and Legendre function expansions, the coefficients are uniquely determined. Therefore, this illustrates the fact that an electromagnetic field as defined earlier is uniquely determined by the value of the tangential E field on any sphere of radius $\rho_1 > \rho_0$.

The spherical-wave expansion program (orthogonality version) given in Appendix D evaluates Eq. (A-10) to obtain $a'_{o^e mn}$ and $b'_{o^e mn}$. That is, it is assumed that the azimuthal Fourier expansion of Eq. (A-6) has already been performed, and the input data to the spherical-wave expansion program are the (tabular) functions $A_{m^e}(\theta)$ and $B_{m^e}(\theta)$. Because successive cases with different values of m may be evaluated, in principle an arbitrary pattern can be expanded. However, the $m = 1$ case is of particular importance, as mentioned previously, which is why the program was written to handle only a single azimuthal component at one time. The odd components have also been neglected to avoid unnecessary complication, since the vast majority of problems may be analyzed using linear polarization. In the case of far-field input data, Eq. (A-4) may be rewritten with the asymptotic form of the Hankel functions, thereby yielding for $\rho_1 \to \infty$

$$a'_{o^e mn} = a_{o^e mn} j^{n+1} \frac{\exp(-jk\rho_1)}{k\rho_1} \tag{A-11a}$$

$$b'_{o^e mn} = b_{o^e mn} j^n \frac{\exp(-jk\rho_1)}{k\rho_1} \tag{A-11b}$$

The scattering program implicitly assumes Eq. (A-11) so it accepts the wave coefficients as output from the spherical-wave expansion program (it is assumed a far-field pattern is input to the spherical-wave expansion program).

Since the solution is known to exist and because it is unique, it is also possible to solve Eq. (A-6) directly. It is well known that if all sources are enclosed in the sphere of radius ρ_0, only modes for which $n \leq k_0$ will make a significant contribution to the field (Refs. 39–41). Therefore, Eq. (A-6) can be solved as a system of linear equations to obtain the coefficients. Again, with only even terms and a single m component, the m and ^e_o subscripts can be

dropped and the matrix equation can be written:

$$\begin{bmatrix} A(\theta_1) \\ B(\theta_1) \\ A(\theta_2) \\ B(\theta_2) \\ \cdot \\ \cdot \\ \cdot \\ A(\theta_N) \\ B(\theta_N) \end{bmatrix} = \pi \begin{bmatrix} F_1^m(\theta_1) & -G_1^m(\theta_1) & F_2^m(\theta_1) \cdots & -G_N^m(\theta_1) \\ G_1^m(\theta_1) & -F_1^m(\theta_1) & G_2^m(\theta_1) \cdots & -F_N^m(\theta_1) \\ F_1^m(\theta_2) & -G_1^m(\theta_2) & F_2^m(\theta_2) \cdots & -G_N^m(\theta_2) \\ G_1^m(\theta_2) & -F_1^m(\theta_2) & G_2^m(\theta_2) \cdots & -F_N^m(\theta_2) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ F_1^m(\theta_N) & -G_1^m(\theta_N) & F_2^m(\theta_N) \cdots & -G_N^m(\theta_N) \\ G_1^m(\theta_N) & -F_1^m(\theta_N) & G_2^m(\theta_N) \cdots & -F_N^m(\theta_N) \end{bmatrix} \begin{bmatrix} a'_1 \\ b'_1 \\ a'_2 \\ b'_2 \\ \cdot \\ \cdot \\ \cdot \\ a'_N \\ b'_N \end{bmatrix} \tag{A-12}$$

where

$$F_n^m(\theta) \equiv \frac{mP_n^m(\cos\theta)}{\sin\theta}$$

$$G_n^m(\theta) \equiv \frac{\partial}{\partial\theta} P_n^m(\cos\theta)$$

$$N \simeq k\rho_0$$

Since $F_n^m(0 \text{ deg}) = G_n^m(0 \text{ deg})$ and $F_n^m(180 \text{ deg}) = -G_n^m(180 \text{ deg})$, the values $\theta = 0$ and 180 deg would make the above matrix singular; therefore, they must not be used as data points. Even excluding these values, the first attempts to invert this system of equations indicated that the matrix was extremely ill-conditioned (on physical grounds it can not be singular if the θ values are distinct and if $\theta = 0$ and 180 deg are excluded). It was subsequently determined that the matrix was ill-conditioned because the data points were concentrated in the region between 0 and 40 deg (where $A(\theta)$ and $B(\theta)$ had significant amplitudes). When the data points were equally spaced between 0 and 180 deg, the matrix was easily inverted. The program that was used to accomplish this, the spherical-wave expansion program (linear equation version), is also presented in Appendix D. When properly used, both programs yielded the same results except for modes containing a negligible fraction of the total power.

The linear equation method is included here primarily because it may be easily generalized to be valid when the fields are known on any surface, whereas the formulation of the orthogonality technique is valid only over the surface of a sphere. Furthermore, the linear equation method provides an interesting numerical check on the other program. However, the orthogonality method is more convenient for a number of reasons: (1) it is faster to perform the numerical integration than to invert the matrix, (2) the number of mode coefficients is not restricted to the number of input θ values, and (3) Parseval's formula can be used to show that a sufficient number of modes have been considered, rather than the well documented but essentially qualitative arguments leading to the $N \simeq k\rho$ relationship.

The equations for obtaining wave coefficients derived here differ slightly from those obtained previously. The main difference is that these equations have been programmed and used to obtain wave expansions of three kinds of patterns—analytical, numerically computed, and experimental. Previous work has been restricted to interesting but idealized cases in which the coefficients were obtained analytically in closed form; e.g., Potter (Ref. 41) found the coefficients for a circularly symmetric optimum illumination pattern; Kennaugh and Ott (Ref. 42) found the wave excited by a plane wave normally incident on a paraboloid; Jones (Ref. 32) gives the coefficients for a plane wave and for an arbitrarily located electric dipole.

Appendix B

Fast Trigonometric Subroutine

The machine-language function subroutine* EXPJX was written for the IBM 7094 computer and returns the complex-valued result $e^{jx} = \cos x + j \sin x$, for real x, in one-half the time required by the IBM library subroutine, and with only slight reduction in accuracy.

A table lookup technique is employed where 64 values of $\sin x_m$ at $x_m = m\,(\pi/128) + (\pi/256)$; $m = 0, 1, \cdots, 63$ are built into the subroutine.

The basic algorithm used is as follows:

(1) $\sin(-x) = -\sin(x)$, $\cos(-x) = \cos(x)$, assume $x \geq 0$.

(2) Write $x = n\,(\pi/128) + r\,(\pi/128)$ where n is an integer, and $1 > r \geq 0$.

(3) Reduce n mod 256 to discard multiples of 2π.

(4) Write $n = 64\,q + m$, where $63 \geq m \geq 0$. Then $q + 1$ is the quadrant of the argument, and m is the index of the tabular value x_m closest to x (reduced to the 1st quadrant).

(5) Define $\Delta x = x - x_m = (r - 1/2)\ \pi/128$; and $|\Delta x| < \pi/265$.

(6) Use $\cos x_m = \sin x_{63-m}$, the expansions

$$\sin(x_m + \Delta x) = \sin x_m \cos \Delta x + \cos x_m \sin \Delta x$$

$$\simeq \sin x_m \left(1 - \frac{\Delta x^2}{2!}\right) + \cos x_m \left(\Delta x - \frac{\Delta x^3}{3!}\right)$$

$$\cos(x_m + \Delta x) = \cos x_m \cos \Delta x - \sin x_m \sin \Delta x$$

$$\simeq \cos x_m \left(1 - \frac{\Delta x^2}{2!}\right) - \sin x_m \left(\Delta x - \frac{\Delta x^3}{3!}\right)$$

and the approximation

$$x - \frac{\Delta x^3}{3!} \simeq \Delta x(1 - 0.16 \times 10^{-4})$$

for $|\Delta x| < (\pi/256)$ to compute $\sin x$ and $\cos x$ in the 1st quadrant.

(7) Use the quadrant index q and the sign of the original argument to obtain $\sin x$ and $\cos x$ for the unreduced (true) argument, and return the complex number $(\cos x, \sin x)$.

The reason for choosing 64 (or in general 2^n) tabular values is that a single multiplication of the input argument $x(128/\pi)$ results in a binary number in which the fractional part is r, the next six higher order bits are m, the next two bits are q, and higher order bits represent multiples of 2π. Therefore, steps (2), (3), and (4) may be accomplished very quickly by one multiplication followed by shifting and masking operations.

The MAP subroutine, which was named EXPJX, requires a total of 167 (decimal) storage locations, compared to 123 locations for the IBM Fortran IV library routine FCSN.

An accuracy and timing test was made in which results from EXPJX and FCSN were compared to the IBM double precision Fortran IV library routine FDSC. The library routine was called twice to obtain values for both $\sin x$ and $\cos x$. Values were computed for 50×10^3 random arguments uniformly distributed over the range $(-5$ to $5\pi)$, with results as shown in Table B-1.

The relative accuracy of EXPJX becomes poor for small values of $\sin x$ or $\cos x$, and for arguments $|x| > 8\pi$ the loss of significance in the reduced argument will increase the errors.

The subroutine of Printout B-1 has been submitted for distribution through SHARE (SHARE library distribution number SDA 3534), and COSMIC (Program NPO 10439).

Table B-1. Performance statistics for fast trigonometric subroutine

Program	Absolute error (rms)	Maximum absolute error	Average speed, μs
EXPJX	8.2×10^{-8}	3.7×10^{-7}	281
FCSN	0.28×10^{-8}	0.13×10^{-7}	571[a]

[a]For both sine and cosine.

*J. Hatfield of JPL Section 314, and H. Thacher of Notre Dame University were associated with the author in the development of this subroutine.

Printout B-1. Computer printout of fast trigonometric subroutine

```
$IBMAP EXPJX   LIST,REF,M94,()OK
       LBL     EXPJX,BEGIN,
       TTL     COMPLEX EXPONENTIAL SUBROUTINE...@EXPJX@
       REM     ***************************************************
       REM     *******                                     *******
       REM     ******* COMPLEX EXPONENTIAL SUBROUTINE...@EXPJX@*******
       REM     *******                                     *******
       REM     ***************************************************
       SPACE   1
       ENTRY   EXPJX
N      SET     64                  SIZE OF SINE-COSINE TABLE
       SPACE   2
       REM     THIS SUBROUTINE COMPUTES
       SPACE   1
       REM                   EXP(J*X)=COS(X)+J*SIN(X)
       SPACE   1
       REM     GIVEN THE FLOATING PT. VALUE, X.
       REM     THE COS(X) AND SIN(X) ARE TABULAR EVALUATED AND @EXPJX@
       REM     RETURNS WITH THEIR RESPECTIVE RESULTS IN THE AC. AND MQ.
       SPACE   3
       REM     PROCEDURE...
       SPACE
       REM       1. COMPUTE Y=(1/DTHETA)*X AND SAVE SIGN OF X
       REM       2. UNFLOAT Y WITH INTEGER IN AC. AND FRACTION IN MQ
       REM       3. UNPACK INTEGER SO THAT QUAD. NO. AND TABLE INDEX
       REM            ARE BITS 28-29 AND 30-35 RESPECTIVELY
       REM       4. COMPUTE DX=ABS(FRACTION OF Y)*DTHETA-DTHETA/2
       REM       5. COMPUTE T1=SIN(DX) AND T2=COS(DX) USING SERIES
       REM            EVALUATION (AT MOST TWO TERMS)
       REM       6. COMPUTE F1=SIN(X@)*T1+COS(X@)*T2
       REM                  F2=COS(X@)*T1-SIN(X@)*T2
       REM            WHERE SIN(X@) AND COS(X@) ARE OBTAINED FROM
       REM            IT@S CORRESPONDING TABLE X@=X-DX
       REM       7. FOR THE APPROPRIATE QUADRANT COMPUTE..
       REM          1ST QUAD. SIN(X)=(SIGN) F1
       REM                    COS(X)=F2
       REM          2ND QUAD. SIN(X)=(SIGN) F2
       REM                    COS(X)=-F1
       REM          3RD QUAD. SIN(X)=-(SIGN) F1
       REM                    COS(X)=-F2
       REM          4TH QUAD. SIN(X)=-(SIGN) F2
       REM                    COS(X)=F1
       REM       8. FLOAT SIN(X) AND COS(X)
       SPACE   3
EXPJX  SXA     EXIT1,4
       SXA     EXIT1+1,1
       SXA     EXIT1+2,2
       SPACE   1
       LDQ*    3,4                 FETCH X (IN RADIANS)
       STQ     SIGN
       FMP     1OVDT               COMPUTE X/DTHETA
       SSP
       UFA     =0233000000000      UNFLOAT IT (I IN ACC., F IN MQ.)
       RQL     8                   SCALE F AT BIT 0 IN MQ.
       LRS     0                   SET SIGN OF F POSITIVE
       ANA     =0377               MASK I FOR QUADRANT AND INDEX NUMBERS
       LGR     6
       PAC     0,1                  XR1=QUADRANT NUMBER
       PXA     0,0
       LGL     6
       PAC     0,2                  XR2=INDEX NUMBER FOR SINE
       PAX     0,4                  XR4=INDEX NUMBER FOR COSINE
       MPY     DT                  COMPUTE F@=F*(PI/2)/64
       SUB     DTOV2
       STO     DX                  SAVE DELTAX=F@-(PI/2)/128
       ARS     1
       CHS
       XCA
       MPY     DX
       ALS     1                   CHANGE SCALE FROM BIT 2 TO BIT 1
```

Printout B-1 (contd)

```
        ADD     =1B1
        STO     T1                  T1=1-DELTAX**2/2          (COS(DELTAX))
        LDQ     SINX,2
        MPY     T1
        STO     F1                  F1=SIN(X@)*T1
        LDQ     COSX,4
        MPY     T2
        ADD     F1
        STO     F1                  F1=SIN(X@)*T1+COS(X@)*T2
        LDQ     SINX,2
        MPY     T2
        STO     F2                  F2=SIN(X@)*T2
        LDQ     COSX,4
        MPY     T1
        SUB     F2
        STO     F2                  F2=COS(X@)*T1-SIN(X@)*T2
        TRA     *+1,1
        TRA     QUAD1
        TRA     QUAD2
        TRA     QUAD3
        TRA     QUAD4
        SPACE   1
QUAD1   CLA     F1
        LDQ     SIGN
        LLS     0
        LDQ     F2
        TRA     EXIT                (SIN(X)=SIGN.F1, COS(X)=F2)
        SPACE   1
QUAD2   LDQ     F2
        CLA     SIGN
        LRS     0
        CLS     F1
        XCA
        TRA     EXIT                (SIN(X)=SIGN.F2, COS(X)=-F1)
        SPACE   1
QUAD3   LDQ     F1
        CLA     SIGN
        CHS
        LRS     0
        CLS     F2
        XCA
        TRA     EXIT                (SIN(X)=-SIGN.F1, COS(X)=-F2)
        SPACE   1
QUAD4   CLA     F2
        LDQ     SIGN
        LLS     0
        CHS
        LDQ     F1                  (SIN(X)=-SIGN.F2, COS(X)=F1)
        SPACE
EXIT    STO     SIN(X)
        CLA     =0202               PACK IN EXPONENT FOR BIT 2 SCALE
        LLS     0                      BUT BE SURE AND RETAIN SIGN OF
        LRS     8                      COS(X)... THE RESULT IS AN
        XCA                            UNNORMALIZED FL. PT. NO. (IN ACC.)
        FAD     =0                  NORMALIZE COS(X) AND SAVE
        STO     COS(X)                 THE FLOATING POINT RESULT
        LDQ     SIN(X)
        CLA     =0202               PACK IN EXPONENT FOR BIT 2 SCALE AND
        LLS     0                      MAKE SURE THE SIGN OF SIN(X) IS
        LRS     8                      RETAINED...THE RESULT IS AN
        XCA                            UNNORMALIZED FL. PT. NO. (IN ACC.)
        FAD     =0                  NORMALIZE SIN(X) AND
        XCA                            PLACE RESULT IN MQ.
        CLA     COS(X)                 PLACE COS(X) IN ACC.
EXIT1   AXT     **,4
        AXT     **,1
        AXT     **,2
        TRA     1,4
        EJECT
DT      DEC     0.024543299981  (PI/2)*(1/64)*0.999984  (SCALED AT BIT 1)
```

Printout B-1 (contd)

```
DTOV2   DEC     0.0122716499B1  (PI/2)*(1/64)/2*0.999984(SCALED AT BIT 1)
1OVDT   DEC     40.7436654          1/DTHETA  (FLOAT. PT.)
DX      PZE     **                  DELTAX
SIGN    PZE     **                  SIGN OF X
T1      PZE     **                  COSINE OF DELTAX     (SCALED AT BIT 1)
T2      EQU     DX                  SINE OF DELTAX       (SCALED AT BIT 1)
F1      PZE     **                  SIN(X)*T1+COS(X)*T2 (SCALED AT BIT 2)
F2      PZE     **                  COS(X)*T1-SIN(X)*T2 (SCALED AT BIT 2)
SIN(X)  PZE     **                  ABSOLUTE VALUE OF SIN(X)
COS(X)  PZE     **                  ABSOLUTE VALUE OF COS(X)
        EJECT
        REM     THE FOLLOWING TABLE CONTAINS FIXED-POINT VALUES
        REM     SCALED AT BIT 1
SINX    NULL
        DEC     0.012271538 3B1
        DEC     0.036807222 9B1
        DEC     0.061320736 3B1
        DEC     0.085797312 3B1
        DEC     0.110222207 3B1
        DEC     0.134580708 5B1
        DEC     0.158858143 3B1
        DEC     0.183039888 0B1
        DEC     0.207111376 2B1
        DEC     0.231058108 3B1
        DEC     0.254865659 6B1
        DEC     0.278519689 4B1
        DEC     0.302005949 3B1
        DEC     0.325310292 2B1
        DEC     0.348418680 3B1
        DEC     0.371317194 0B1
        DEC     0.393992040 1B1
        DEC     0.416429560 1B1
        DEC     0.438616238 5B1
        DEC     0.460538711 0B1
        DEC     0.482183772 1B1
        DEC     0.503538383 7B1
        DEC     0.524589682 7B1
        DEC     0.545324988 4B1
        DEC     0.565731810 8B1
        DEC     0.585797857 5B1
        DEC     0.605511041 4B1
        DEC     0.624859488 1B1
        DEC     0.643831542 9B1
        DEC     0.662415777 6B1
        DEC     0.680600997 8B1
        DEC     0.698376249 4B1
        DEC     0.715730825 3B1
        DEC     0.732654271 7B1
        DEC     0.749136394 5B1
        DEC     0.765167265 6B1
        DEC     0.780737228 6B1
        DEC     0.795836904 6B1
        DEC     0.810457198 3B1
        DEC     0.824589302 8B1
        DEC     0.838224705 6B1
        DEC     0.851355193 1B1
        DEC     0.863972856 1B1
        DEC     0.876070094 2B1
        DEC     0.887639620 4B1
        DEC     0.898674465 7B1
        DEC     0.909167983 1B1
        DEC     0.919113851 7B1
        DEC     0.928506080 5B1
        DEC     0.937339011 9B1
        DEC     0.945607325 4B1
        DEC     0.953306040 4B1
        DEC     0.960430519 4B1
        DEC     0.966976471 0B1
        DEC     0.972939952 2B1
        DEC     0.978317370 7B1
```

Printout B-1 (contd)

```
        DEC     0.9831054874B1
        DEC     0.9873014182B1
        DEC     0.9909026354B1
        DEC     0.9939069700B1
        DEC     0.9963126122B1
        DEC     0.9981181129B1
        DEC     0.9993223846B1
        DEC     0.9999247018B1
COSX    EQU     SINX+N-1
        END
```

Appendix C

Tilted Reflector Program

The tilted reflector program was written for an IBM 1620 computer and outputs a deck of cards containing the quantities (see Eq. 12):

$$a_0(\theta), \qquad a_1(\theta), \qquad a_2(\theta),$$

$$\frac{\partial a_0}{\partial \theta}, \qquad \frac{\partial a_1}{\partial \theta}, \qquad \frac{\partial a_2}{\partial \theta}$$

The input data are tabular values $\rho'(\theta')$ describing a surface of revolution, as illustrated in Fig. C-1. Also specified are the translations to the new origin XF and ZF; a coordinate rotation is specified by the intersection point ZROT.

The procedure used is as follows:

(1) By the use of a straightforward coordinate transformation, the data

$$\rho'(\theta',0\text{ deg}) = \rho'(\theta',90\text{ deg}) = \rho'(\theta',180\text{ deg}) = \rho'(\theta')$$

are mapped into the new system for each input θ' value. Note that the cuts $\phi' = 0$ or 180 deg map into $\phi = 0$ or 180 deg, but in general $\phi' = 90$ deg maps into a cut which will be identified by $\phi_{90}(\theta)$. The result of this step is data

$$\rho(\theta,0\text{ deg}), \qquad \rho(\theta,\phi_{90}), \qquad \rho(\theta,180\text{ deg})$$

that are known at a set of θ values that is in general different for each ϕ cut, and different from the set of desired output values. The remaining steps (2) through (4) are performed for each desired output value θ_0.

(2) A cubic interpolation polynomial in θ is fitted locally at four points to the functions $\rho(\theta,0\text{ deg})$, $\rho(\theta,\phi_{90})$, $\rho(\theta,180\text{ deg})$ and $\phi_{90}(\theta)$. These polynomials and their derivatives are evaluated at $\theta = \theta_0$ to yield $\phi_{90}(\theta_0)$ and $\partial\phi_{90}/\partial\theta$ at $\theta = \theta_0$; and $\rho(\theta_0,0\text{ deg})$, $\rho(\theta_0,\phi_{90})$, $\rho(\theta_0,180\text{ deg})$ and $d\rho/d\theta$ at $\phi = 0$, ϕ_{90}, and 180 deg.

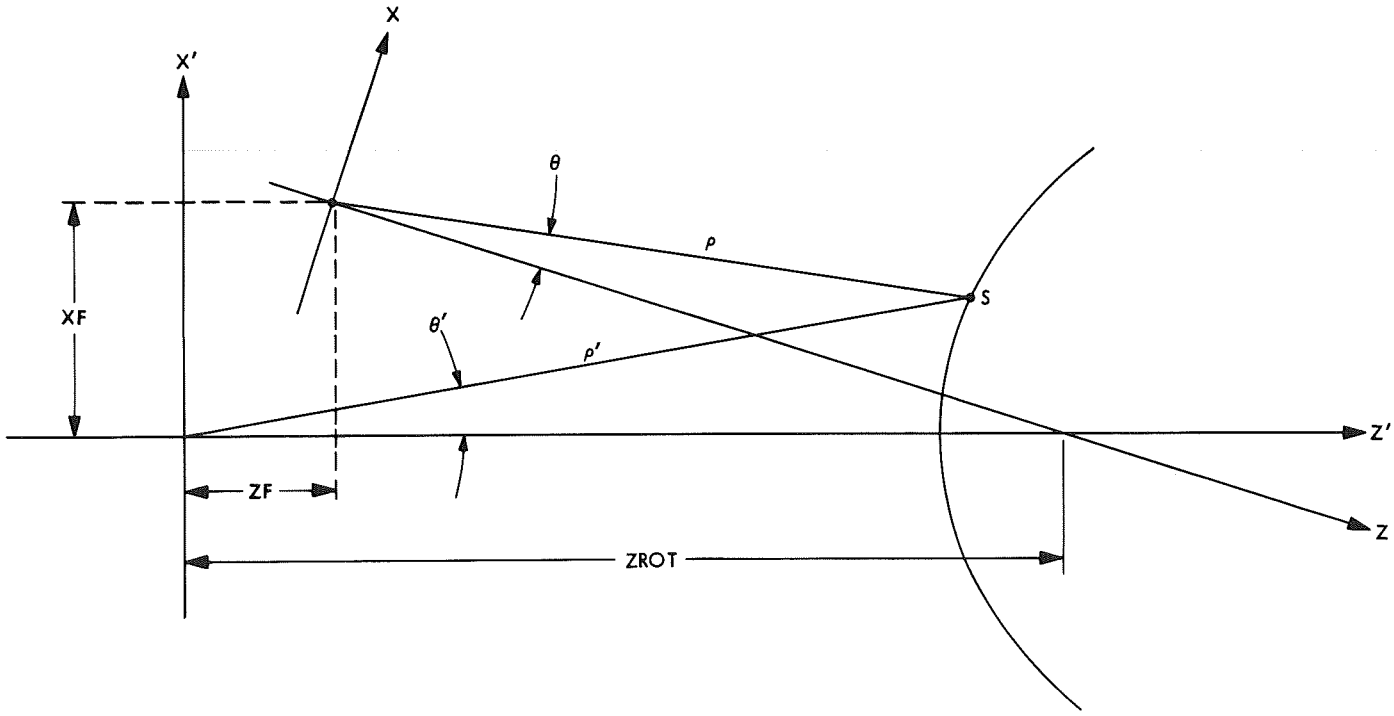


Fig. C-1. Reflector coordinate transformation

(3) Given the assumed form

$$\rho(\theta,\phi) = a_0(\theta) + a_1(\theta)\cos\phi + a_2(\theta)\cos 2\phi$$

it follows that

$$\begin{bmatrix} \rho(\theta_0,0) \\ \rho(\theta_0,\phi_{90}) \\ \rho(\theta_0,180) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & \cos\phi_{90} & \cos 2\phi_{90} \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} a_0(\theta_0) \\ a_1(\theta_0) \\ a_2(\theta_0) \end{bmatrix}$$

By the use of $\phi_{90}(\theta_0)$ from step (2), the program then inverts the matrix and obtains $a_0(\theta_0)$, $a_1(\theta_0)$, and $a_2(\theta_0)$.

For the matrix to be well-conditioned, it is sufficient that $\phi_{90} \simeq 90$ deg. If ϕ_{90} lies outside the range of 60 to 120 deg the program sets $a_2(\theta) = 0$ and solves the two-dimensional matrix problem for $a_0(\theta)$ and $a_1(\theta)$.

(4) Also, from the assumed form,

$$\frac{\partial\rho}{\partial\theta} = \frac{\partial a_0}{\partial\theta} + \frac{\partial a_1}{\partial\theta}\cos\phi + \frac{\partial a_2}{\partial\theta}\cos 2\phi$$

$$\frac{\partial\rho}{\partial\phi} = -a_1(\theta)\sin\phi - 2a_2(\theta)\sin 2\phi$$

By the use of the relationship

$$\frac{\partial\rho}{\partial\theta} = \frac{d\rho}{d\theta} - \frac{\partial\rho}{\partial\phi}\frac{\partial\phi}{\partial\theta}$$

$$= \frac{d\rho}{d\theta} + [a_1(\theta)\sin\phi + 2a_2(\theta)\sin 2\phi]\frac{\partial\phi}{\partial\theta}$$

and the values of $d\rho/d\theta$ and $\partial\phi_{90}/\partial\theta$ at $\theta = \theta_0$ obtained in step (2), values of $\partial\rho/\partial\theta$ are obtained at $\phi = 0$ deg, ϕ_{90}, and 180 deg. Then the same matrix inverse obtained in step (3) is used to yield $\partial a_0/\partial\theta$, $\partial a_1/\partial\theta$ and $\partial a_2/\partial\theta$.

The input data for the tilted reflector program are shown in Table C-1; the parameters are defined below:

TITLE any alphanumeric statement

XF, ZF, ZROT see Fig. C-1

T, R θ',ρ' (Note that θ' must be monotonically decreasing with its index.)

9999.0 any number > 400 (A last card indicator. NMAX must be ≤ 200. If the input consists of more than 200 cards, the program will type out an error message; the program can be reset to read title card.)

NMAX number of θ values desired

TOUT desired θ values

The program prints out the input data (ρ',θ') and transformed data (ρ,θ) at $\phi = 0$, 180 deg, and ϕ_{90}. Then ρ and $\partial\rho/\partial\theta$ are printed out at the desired θ values (Printout C-1).

The Fourier coefficients and their derivatives are punched but not printed.

Table C-1. Input data for the tilted reflector program

Card	Parameters	Format
1	TITLE	20A4
2	XF ZF ZROT	8F10.0
3	T(1) R(1)	20X, 2F10.0
·	T(2) R(2)	· ·
·	· ·	· ·
	· ·	
·	· ·	· ·
NMAX + 2	T(NMAX) R(NMAX)	20X, 2F10.0
NMAX + 3	9999.0	20X, F10.0
NMAX + 4	MMAX	1015
NMAX + 5	TOUT(1) TOUT(2) · · · TOUT(8)	8F10.0
		·
·	TOUT(9) TOUT(10) · · ·	·
·		·
·	· · · TOUT(MMAX)	8F10.0

Printout C-1. Computer printout of the tilted reflector program

```
C      TILTED REFLECTOR PROGRAM
C
       DIMENSION T(200),R(200)
       DIMENSION TITLE(20)
       DIMENSION RO(200),TO(200),RE(200),TE(200),RN(200),TN(200),PN(200)
       DIMENSION TOUT(70)
       DTR=0.017453293
     1 READ 1001,TITLE
       PUNCH 1001,TITLE
       PRINT 2001,TITLE
       READ 1002,XF,ZF,ZROT
       PRINT 2002,XF,ZF,ZROT
       DZ=ZROT-ZF
       CALL ATANXY(DZ,XF,BETA)
       BEDEG=BETA/DTR
       PRINT 2003,BEDEG
       SIB=SINF(BETA)
       COB=COSF(BETA)
C
C      READ IN SHAPED SUBREFLECTOR
C
       N=0
    10 N=N+1
       READ 1003,T(N),R(N)
       T(N)=T(N)*DTR
       IF(T(N)-4.0)15,20,20
    15 IF(N-200)10,99,99
    20 NMAX=N-1
       PRINT 2004,NMAX
C
C      TRANSFORM DATA TO NEW SYSTEM
C
       CALL TRANQ(R,T,RO,TO,PN,0.0, 1.0,SIB,COB,XF,ZF,NMAX)
       DO 24 N=1,NMAX
    24 TO(N)=TO(N)*COSF(PN(N))
       CALL TRANQ(R,T,RE,TE,PN,0.0,-1.0,SIB,COB,XF,ZF,NMAX)
       DO 25 N=1,NMAX
    25 TE(N)=-TE(N)*COSF(PN(N))
       CALL TRANQ(R,T,RN,TN,PN,1.0, 0.0,SIB,COB,XF,ZF,NMAX)
       PRINT 2008
       DO 26 N=1,NMAX
       TOO=TO(N)/DTR
       TNN=TN(N)/DTR
       TEE=TE(N)/DTR
       TTT=T(N)/DTR
       PNN=PN(N)/DTR
    26 PRINT 2009,TTT,R(N),TOO,RO(N),TEE,RE(N),TNN,RN(N),PNN
C
C      READ DESIRED THETA VALUES
C
       READ 1004,MMAX
       IF(MMAX- 75)29,29,99
    29 READ 1002,(TOUT(N),N=1,MMAX)
       NLASTO=NMAX-2
       NLASTE=NMAX-2
       NLASTN=NMAX-2
       PRINT 2007
       DO 30 MM=1,MMAX
       TTO=TOUT(MM)*DTR
C
C      INTERPOLATE TO DESIRED THETA VALUES
C
       CALL QBURP(TO,RO,TTO,RRO,DRO,NLASTO,NMAX)
       CALL QBURP(TE,RE,TTO,RRE,DRE,NLASTE,NMAX)
       CALL QBURP(TN,RN,TTO,RRN,DRN,NLASTN,NMAX)
       CALL QBURP(TN,PN,TTO,PPN,DPN,NLASTN,NMAX)
       PNN=PPN/DTR
C
C      FIND FOURIER COEIFFICIENTS
C
```

Printout C-1 (contd)

```
      CALL FKQSF(RRO,RRE,RRN,PPN,A0,A1,A2)
      DRN=DRN+(A1*SINF(PPN)+2.0*A2*SINF(2.0*PPN))*DPN
      PRINT 2005,TOUT(MM),RRO,DRO,RRE,DRE,RRN,DRN,PNN,DPN
      CALL FKQSF(DRO,DRE,DRN,PPN,DA0,DA1,DA2)
      PUNCH 2006,TOUT(MM),A0,A1,A2,DA0,DA1,DA2
   30 CONTINUE
      GO TO 1
C
   99 TYPE 2010
      PAUSE
      GO TO 1
C
 1001 FORMAT(20A4)
 1002 FORMAT(8F10.0)
 1003 FORMAT(20X,2F10.0)
 1004 FORMAT(10I5)
 2001 FORMAT(1H1,20A4)
 2002 FORMAT(28H   COORDINATES OF NEW ORIGIN/
     *7H      X=,E15.8/
     *7H      Z=,E15.8/
     *26H   POINT OF ROTATION ZROT=,E15.8)
 2003 FORMAT(23H   ROTATION ANGLE BETA=,F8.4,9H  DEGREES)
 2004 FORMAT(31H   NUMBER OF INPUT POINTS NMAX=,I4)
 2005 FORMAT(F10.5,3(F14.5,F12.5),2F11.5)
 2006 FORMAT(7F10.4)
 2007 FORMAT(1H1,47H   R AND ITS PARTIAL DERIVATIVE WITH RESPECT TO,
     *31H THETA AT DESIRED OUTPUT POINTS//
     *20X5HPHI=0,21X7HPHI=180,21X16HPHI AS TABULATED/
     *10H   THETA  ,3(8X,1HR,10X,7HDR/DT  ),4X3HPHI,7X7HDPHI/DT)
 2008 FORMAT(//52H   SURFACE DATA IN INPUT AND TRANSFORMED COORDINATES//
     *7X,10HINPUT DATA,11X,15HPHI= 0  DEGREES,10X,15HPHI=180 DEGREES,15X
     *,16HPHI AS TABULATED/
     *4(8H   THETA,8X,1HR,8X),4H PHI)
 2009 FORMAT(F10.5,F12.5,3(F13.5,F12.5),F11.5)
 2010 FORMAT(41H   INPUT DATA EXCEEDS DIMENSION STATEMENT/
     *42H   RELOAD DATA AND PUSH START TO TRY AGAIN)
      END

      SUBROUTINE QBURP(T,R,TT,RR,DR,NLAST,NMAX)
      DIMENSION T(2),R(2)
      DIMENSION B(4,5),A(4)
C
C     THIS SUBROUTINE NUMERICALLY INTERPOLATES AND DIFFERENTIATES THE
C     FUNCTION R(T) BY FITTING A CUBIC  TO R(T) AT FOUR POINTS
C
C     FIND NLAST SUCH THAT T(NLAST-1) AND T(NLAST) STRADDLE THE
C     DESIRED VALUE TT.  IT IS ASSUMED THAT THE TABULAR VALUES OF T
C     DECREASE WITH INCREASING N
    9 IF(T(NLAST)-TT)10,10,20
   10 IF(NLAST-2)20,20,15
   15 NLAST=NLAST-1
      GO TO 9
   20 CONTINUE
C
C     SOLVE THE SYSTEM OF FOUR EQUATIONS TO DETERMINE THE COEIFFICIENTS
C
C     ESTABLISH MATRIX
      DO 30 I=1,4
      NK=NLAST+I-2
      B(I,5)=R(NK)
      DO 30 J=1,4
      JJ=J-1
   30 B(I,J)=T(NK)**JJ
C
C     DIAGONALIZE SYSTEM
      DO 50 K=1,3
      KK=K+1
      DO 50 I=KK,4
      D=B(I,K)/B(K,K)
      DO 50 L=K,5
   50 B(I,L)=B(I,L)-D*B(K,L)
```

Printout C-1 (contd)

```
C
C       SOLVE BY BACK SUBSTITUTION
        A(4)=B(4,5)/B(4,4)
        DO 70 II=1,3
        I=4-II
        L=5-II
        DO 60 J=L,4
   60 B(I,5)=B(I,5)-B(I,J)*A(J)
   70 A(I)=B(I,5)/B(I,I)
C
C       EVALUATE CUBIC AND ITS DERIVATIVE AT INTERPOLATION POINT TT
C
        RR=A(1)+A(2)*TT+A(3)*TT*TT+A(4)*TT*TT*TT
        DR=A(2)+2.0*A(3)*TT+3.0*A(4)*TT*TT
        RETURN
        END

        SUBROUTINE FKQSF(O,E,FN,P,A0,A1,A2)
C
C       THIS SUBROUTINE COMPUTES THE COEIFFICIENTS FOR THE TRUNCATED
C       FOURIER SERIES R(THETA)=A0+A1*COS(THETA)+A2*COS(2*THETA)
C
C       O=R(ZERO RADIANS)
C       E=R(PI RADIANS)
C       FN=R(P RADIANS) WHERE P SHOULD LIE BETWEEN 1.05 AND 2.09
C
        IF(P-1.05)100,20,20
   20 IF(2.09-P)100,200,200
  100 A0=(O+E)/2.0
        A1=(O-E)/2.0
        A2=0.
        RETURN
  200 C=COSF(P)
        C2=COSF(2.0*P)
        D=2.0*(C2-1.0)
        A0=(C+C2)*O-2.0*FN+(C2-C)*E
        A0=A0/D
        A1=(C2-1.0)*O+(-C2+1.0)*E
        A1=A1/D
        A2=(-C-1.0)*O+2.0*FN+(C-1.0)*E
        A2=A2/D
        RETURN
        END


        SUBROUTINE TRANQ(R,T,RR,TT,PP,SI,CO,SIB,COB,XF,ZF,NMAX)
C
C       THIS SUBROUTINE PERFORMS A COORDINATE TRANSLATION AND ROTATION
C
        DIMENSION R(2),T(2),RR(2),TT(2),PP(2)
C
C       CONVERT TO RECTANGULAR COORDINATES
        DO 10 N=1,NMAX
        X=R(N)*SINF(T(N))*CO
        Y=R(N)*SINF(T(N))*SI
        Z=R(N)*COSF(T(N))
C
C       PERFORM TRANSLATION
        X=X-XF
        Z=Z-ZF
C
C       PERFORM ROTATION
        D=X
        X=X*COB+Z*SIB
        Z=-D*SIB+Z*COB
C
C       CONVERT TO POLAR COORDINATES IN NEW SYSTEM
        RR(N)=SQRTF(X*X+Y*Y+Z*Z)
        D=SQRTF(X*X+Y*Y)
        CALL ATANXY(Z,D,TT(N))
```

Printout C-1 (contd)

```
   10 CALL ATANXY(X,Y,PP(N))
      RETURN
      END

      SUBROUTINE ATANXY(X,Y,PHI)
      PI=3.1415927
      IF(X)10,20,30
   10 PHI=ATANF(Y/X)+PI
      GO TO 40
   20 IF(Y)21,22,23
   21 PHI=-PI/2.0
      GO TO 50
   22 PHI=0.0
      GO TO 50
   23 PHI=PI/2.0
      GO TO 50
   30 PHI=ATANF(Y/X)
      GO TO 50
   40 IF(PHI-PI)50,50,41
   41 PHI=PHI-2.0*PI
   50 RETURN
      END
```

Appendix D

Scattering and Spherical-Wave Expansion Programs

The scattering program consists of the following programs* (see Fig. D-1):

MAIN	asymmetrical scattering program
FIELDS	subroutine that provides $\mathbf{H}_i$ on surface (see Section IV)
SURF	subroutine that provides ρ and its derivatives (see Section III)
SETUP	subroutine that establishes integration and output grids, and tables of precomputed trigonometric functions
PATHL	subroutine that established the path length term, Eq. (17)
FINT	numerical integration subroutine (see Section V)
LEGEND	Legendre functions subroutine
SPHANK	spherical Hankel function subroutine
EXPJX	fast trigonometric subroutine (see Appendix B)
VECTOR	subroutine that converts from rectangular to polar coordinates
ADJUST	subroutine that normalizes angles to the range -180 to 180 deg
PRTIM	subroutine that prints out execution times

The spherical-wave expansion program (orthogonality method) consists of the following subprograms (see Fig. D-2):

MAIN	spherical-wave expansion program (orthogonality method)
MULT	a general purpose matrix multiplication subroutine
LEGEND	Legendre function subroutine
VECTOR	subroutine that converts from rectangular to polar coordinates
PRTIM	subroutine that prints out execution times

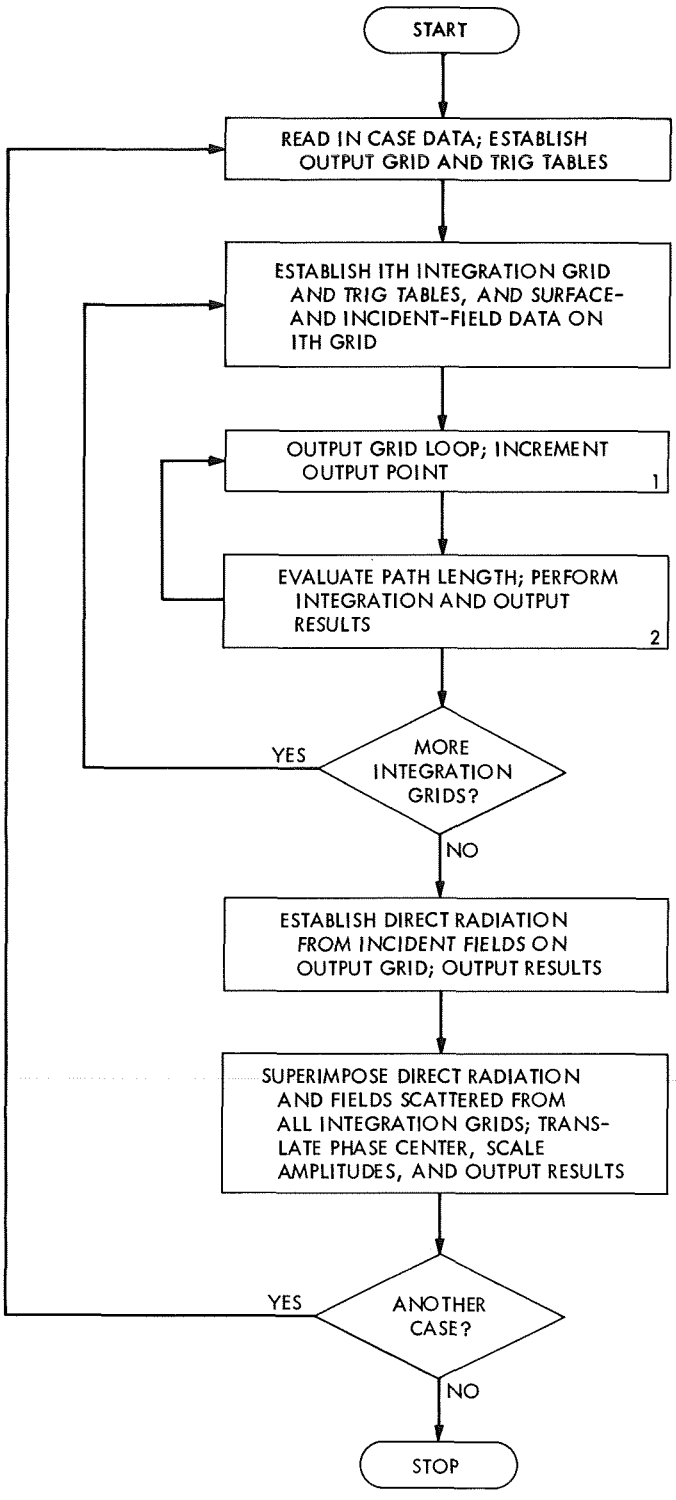
*All the programs presented in this section, except SOLVE and PRTIM, were developed and programmed by the author. The matrix inversion package SOLVE was developed by Richard Hanson of the Computation and Analysis Section of JPL; PRTIM, which prints out the actual computation time during program execution, was developed by Charles Lawson, also of the Computation and Analysis Section.

The spherical-wave expansion program (linear equation method) consists of the following subprograms (see Fig. D-2):

MAIN	spherical-wave expansion program (linear equation method)
SOLVE	a package of subroutines to solve a set of linear equations
LEGEND	Legendre function subroutine
PRTIM	subroutine that prints out execution times

The input data required by the main scattering program are shown in Table D-1; the parameters are as follows:

TITLE	any alphanumeric statement
PC	propagation constant, $2\pi/\lambda$
XT, YT, ZT	translations to reference point for phase pattern of output data (normally the expected phase center of the scattered pattern)
SCALE	a scale factor for the output fields; if this is left blank or if input is zero, program sets SCALE = 1.0
JMAX	number of Θ values desired, JMAX ≤ 181
TT1	initial Θ value
DTT	increment for Θ values, DTT ≥ 0
TT(1), TT(2), $\cdots$	explicit list of Θ values
KMAX	number of Φ values desired in output, KMAX ≤ 8
PP1	initial Φ value
DPP	increment for Φ values, DPP ≥ 0
PP(1), PP(2), $\cdots$	explicit list of Φ values
NG1	number of integration grids, NG1 ≤ 5
MM(I)	number of θ values in the Ith integration grid, MM ≤ 15
T1	initial θ value
DT	increment for θ values, DT ≥ 0



1 THIS IS A JMAX × KMAX LOOP

2 THIS STEP INTERNALLY CONTAINS SEVERAL MMAX × NMAX LOOPS

Fig. D-1. Block diagram of scattering program

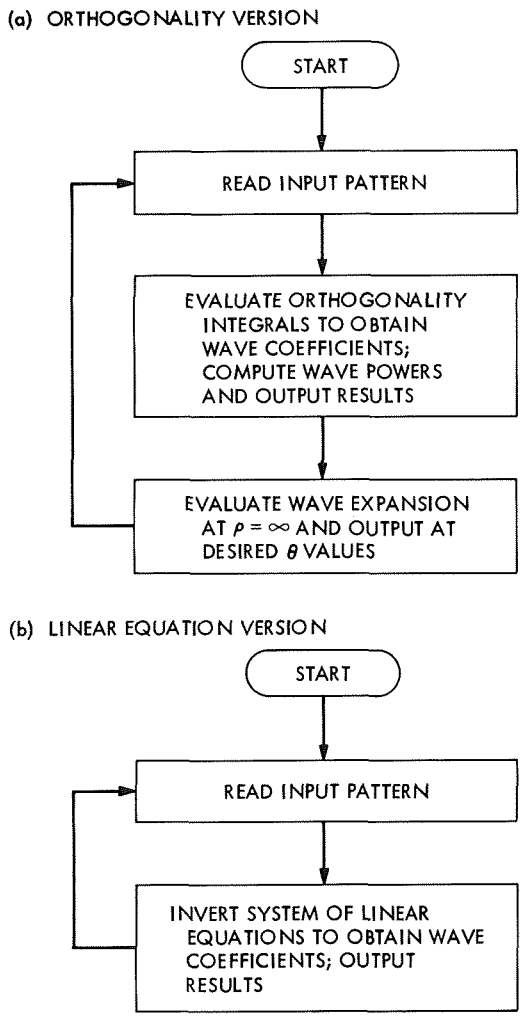


Fig. D-2. Block diagram of spherical-wave expansion program

T(I,1), T(I,2), ···	explicit list of θ values for Ith grid
NN(I)	number of ϕ values in the Ith integration grid, NN $\leq$ 91
P1	initial ϕ value
DP	increment for ϕ values
P(I,1), P(I,2), ···	explicit list of ϕ values for Ith grid
(LAST)	a last case indicator; punched as shown in columns 1 through 6

In all cases, a set of angle values may be specified by an initial value and a positive increment, or by an explicit list of values (not necessarily equally spaced). All angles are in degrees. If a positive increment is given, the data cards containing explicit values *must be omitted.* If

Table D-1. Input for MAIN scattering program

Card	Parameters	Format
1	TITLE	13A6
2	PC XT YT ZT SCALE	5F10.0
3	JMAX TT1 DTT	I5,2F10.0
·	TT(1) TT(2) · · ·	8F10.0
·	· · · TT(JMAX)	8F10.0
·	KMAX PP1 DPP	I5,2F10.0
	PP(1) PP(2) · · ·	8F10.0
	· · · PP(KMAX)	8F10.0
	NG1	I5
	MM(1) T1 DT	I5,2F10.0
	T(1,1) T(1,2) · · ·	8F10.0
	· · · T(1,MM(1))	8F10.0
	NN(1) P1 DP	I5,2F10.0
	P(1,1) P(1,2) · · ·	8F10.0
	· · · P(1,NN(1))	8F10.0
	·	
	·	
	·	
	MM(NG1) T1 DT	I5,2F10.0
	T(NG1,1) T(NG1,2) · · ·	8F10.0
	· · · T(NG1,MM(NG1))	8F10.0
	NN(NG1) P1 DP	I5,2F10.0
	P(NG1,1) P(NG1,2) · · ·	8F10.0
	· · · P(NG1,NN(NG1))	8F10.0
	[Input for FIELDS Subroutine]	—
	[Input for SURF Subroutine]	—
	TITLE	
	PC XT YT ZT SCALE	
	·	
	·	
	·	
	(Data for further cases, same as above)	
	·	
	·	
(Last)	(LAST)	13A6

the data cards containing explicit values are included, *the increment must be written in as identically zero.*

The input data required by the spherical-wave version FIELDS subroutine are shown in Table D-2; the parameters are as follows:

TITLE any alphanumeric statement

LMAX maximum mode order, ≤ 60

MCOMP order of azimuthal variation

Table D-2. Input for spherical-wave FIELDS subroutine

Card	Parameters	Format
1	TITLE	13A6
2	TITLE	13A6
3	LMAX MCOMP	2I5
4	1 A(1,1) A(1,2) B(1,1) B(1,2)	I5,2E17.8, 2X,2E17.8
·	·	·
·	·	·
·	·	·
LMAX + 2	LMAX A(LMAX,1) A(LMAX,2) B(LMAX,1) B(LMAX,2)	·

A(N,1), A(N,2) real and imaginary parts of $TE_{MCOMP,N}$ wave coefficient (see Appendix A and following program)

B(N,1), B(N,2) real and imaginary parts of $TM_{MCOMP,N}$ wave coefficient (see Appendix A and following program)

The input data for the far-field version FIELDS subroutine are shown in Table D-3; the parameters are as follows:

TITLE any alphanumeric statement

JMAX not used

JO not used

JIN number of input points, JIN ≤ 181

IC1 ≤ 0 for input in dB; > 0 for input in V

IC2 > 0 to set phase pattern zero everywhere

MCOMP order of azimuthal variation

PSI polar angle θ (see Fig. 1)

E $|E_\theta(\theta,90 \text{ deg})|$, in V or dB

EP argument $\{E_\theta(\theta,90 \text{ deg})\}$, in deg } see Eq. (14)

H $|E_\phi(\theta,0 \text{ deg})|$, in V or dB } see Eq. (14)

HP argument $\{E_\phi(\theta,0 \text{ deg})\}$, in deg

The input data for the SURF subroutine are shown in Table D-4; the parameters are as follows:

TITLE any alphanumeric statement

TIN input θ value (see note below)

Table D-3. Input for far-field FIELDS subroutine

Card	Parameters	Format
1	TITLE	72H
2	JMAX JO JIN IC1 IC2 MCOMP	615
3	PSI(1) E(1) EP(1) H(1) HP(1)	5F10.6
.	.	.
.	.	.
.	.	.
JIN + 2	PSI(JIN) E(JIN) EP(JIN) H(JIN) HP(JIN)	5F10.6

A0, A1, A2 a_0, a_1, a_2 (see Eqs. 12 and 13, and Appendix C)

DA0, DA1, DA2 $\frac{\partial a_0}{\partial \theta}$, $\frac{\partial a_1}{\partial \theta}$, $\frac{\partial a_2}{\partial \theta}$ (see Eqs. 12 and 13, and Appendix C)

The TIN values must agree with the specified integration grid values. (If they disagree, an error message is printed but the program continues to run.) The values for each integration grid are stacked in order.

Computer time is nearly proportional to the quantity

$$P = \sum_{\substack{\text{NG1 integration} \\ \text{grids}}} \text{NMAX} \times \text{MMAX}$$

For an IBM 7094, the time can vary by a factor of 2 depending on which model is being used; time for a slower model is given by

$$t = P \times [\text{JMAX} \times \text{KMAX}]\ 8.4 \times 10^{-5} \text{ min}$$

plus

$$t = P \times [\text{LMAX}]^2\, 5 \times 10^{-6} \text{ min}$$

if the spherical-wave version FIELDS subroutine is used.

The input data required by the spherical-wave expansion program (orthogonality version) are shown in Table D-5. All of these parameters have been previously defined, except for JMAXO $= 180/\Delta\theta + 1$, where $\Delta\theta$ is the desired output increment, and JOUT is the number of output values starting with $\theta = 0$. The program evaluates the spherical-wave expansion at $\rho = \infty$ and prints and punches the results at a set of output values determined by these parameters.

Table D-4. Input for SURF subroutine

Card	Parameters	Format
1	TITLE	12A6
2	TIN(1) A0(1) A1(1) A2(1) DA0(1) DA1(1) DA2(1)	7F10.4
.	.	.
.	TIN(2) A0(2) A1(2) A2(2) DA0(2) DA1(2) DA2(2)	.
.	.	.
	.	
	.	
	.	
	TIN(MM) A0(MM) A1(MM) A2(MM) DA0(MM) DA1(MM) DA2(MM)	7F10.4
	TIN(1) A0(1) A1(1) · · ·	
	.	
	.	
	.	

The input data required by the spherical-wave expansion program (linear equation version) are identical except that the first two cards and the last card are omitted.

Both of these programs use less than 1 min of computer time to evaluate 60 TE and 60 TM wave coefficients.

Two computer printouts are included in this appendix: the first (Printout D-1) is a computer printout of the asymmetrical scattering program; the second (Printout D-2) is a printout of the spherical-wave expansion program.

Table D-5. Input for spherical-wave expansion program

Card	Parameters	Format
1	TITLE	13A6
2	MCOMP LMAX	215
3	TITLE	
4	JMAX JO JIN IC1 IC2 MCOMP	615
5	PSI(1) E(1) EP(1) H(1) HP(1)	5F10.0
.	.	.
.	.	.
.	.	.
JIN + 4	PSI(JIN) E(JIN) EP(JIN) H(JIN) HP(JIN)	5F10.0
JIN + 5	JMAXO JOUT	215

Printout D-1. Computer printout of asymmetrical scattering program

(a) Main program

```
$IBFTC MAIN
C      ASYMMETRICAL SCATTERING PROGRAM
C      PROGRAM COMPUTES SCATTERING FOR PERFECTLY CONDUCTING SMOOTH
C      SURFACE OF ARBITRARY SHAPE  A. LUDWIG  1-15-69
C
C
C
       COMMON/GRID1/SIT(15),COT(15),SIP(181),COP(181),T(15),P(181)
       COMMON/GRID2/SITT(181),COTT(181),SIPP(  8),COPP(  8),TT(181),PP(8)
       DIMENSION TITLE(13)
       DIMENSION F( 15, 91),FT( 15, 91),FP( 15, 91),GAM( 15, 91)
       INTEGER TITLE
       COMPLEX HR( 15, 91),HT( 15, 91),HP( 15, 91)
       COMPLEX ETT(181, 8),EPP(181, 8)
       COMPLEX ETTO,EPPO
       COMPLEX A( 15, 91,3)
       COMPLEX STOT(3)
       COMPLEX T1,T2,T3
       COMPLEX TET,TEP
       EQUIVALENCE (A(1,1,1),HR),(A(1,1,2),HT),(A(1,1,3),HP)
       EQUIVALENCE (GAM,FT)
       DATA LC/6H(LAST)/
C      START TIMING ROUTINE
       CALL PRTIM1
       READ(5,1001)TITLE
     1 WRITE(6,2001)TITLE
       READ(5,1002)PC,XT,YT,ZT,SCALE
       IF(SCALE)22,21,22
    21 SCALE=1.0
    22 CONTINUE
       WRITE(6,2003)PC
C
C      READ IN GRID DATA AND ESTABLISH OUTPUT GRID AND FIRST INTEGRATION
C      GRID
       CALL SETUP(NG1,I,JMAX,KMAX,MMAX,NMAX)
       CALL PRTIM2
       WRITE(6,2002)TITLE
C
C      READ IN EXCITATION FIELD DATA
       CALL FIELDS
       CALL PRTIM2
C
C      BEGIN LOOP FOR INTEGRATION GRID SEGMENTS
       DO 90 J=1,JMAX
       DO 90 K=1,KMAX
       ETT(J,K)=(0.0,0.0)
    90 EPP(J,K)=(0.0,0.0)
   100 WRITE(6,2002)TITLE
C
C      ESTABLISH SURFACE PARAMETERS ON INTEGRATION GRID
       CALL SURF(MMAX,NMAX,F,FT,FP)
       DO 410 M=1,MMAX
       DO 410 N=1,NMAX
   410 F(M,N)=PC*F(M,N)
C      ESTABLISH EXCITATION FIELDS ON INTEGRATION GRID
       CALL FIELD1(MMAX,NMAX,HR,HT,HP,F)
C      COMBINE SURFACE AND FIELD DATA TO DETERMINE COMPLEX VECTOR A
       DO 400 M=1,MMAX
       DO 400 N=1,NMAX
       T1=FT(M,N)*SIT(M)*HP(M,N)-FP(M,N)*HT(M,N)
       T2=FP(M,N)*HR(M,N)+F(M,N)*SIT(M)*HP(M,N)/PC
       T3=-FT(M,N)*SIT(M)*HR(M,N)-F(M,N)*SIT(M)*HT(M,N)/PC
       A(M,N,1)=T1*SIT(M)*COP(N)+T2*COT(M)*COP(N)-T3*SIP(N)
       A(M,N,2)=T1*SIT(M)*SIP(N)+T2*COT(M)*SIP(N)+T3*COP(N)
       A(M,N,3)=T1*COT(M)-T2*SIT(M)
   400 CONTINUE
       WRITE(6,2007)I
       CALL PRTIM2
C
C      BEGIN OUTPUT GRID LOOP
```

Printout D-1 (contd)

```
      WRITE(6,2006)I
C
      DO 500 K=1,KMAX
      TO=PP(K)/0.017453293
      WRITE(6,2011)TO
      DO 500 J=1,JMAX
C
C     ESTABLISH PHASE/PATH LENGTH PARAMETER ON INTEGRATION GRID
      CALL PATHL(F,J,K,MMAX,NMAX,GAM)
C
C     PERFORM INTEGRATION
      CALL FINT(T,P,A,GAM,MMAX-1,NMAX-1,STOT)
C
C     CONVERT RESULT TO POLAR COORDINATES AND SUPERINPOSE FIELDS
C
      ETTO=COTT(J)*(STOT(1)*COPP(K)+STOT(2)*SIPP(K))-STOT(3)*SITT(J)
      EPPO=STOT(2)*COPP(K)-STOT(1)*SIPP(K)
      ETTO=-(0.0,1.0)*PC/6.2831854*ETTO
      EPPO=-(0.0,1.0)*PC/6.2831854*EPPO
      TO=TT(J)/0.017453293
      A1=REAL(ETTO)
      A2=AIMAG(ETTO)
      A3=REAL(EPPO)
      A4=AIMAG(EPPO)
      CALL VECTOR(A1,A2,ETAMP,ETPHI)
      CALL VECTOR(A3,A4,EPAMP,EPPHI)
      WRITE(6,2012)TO,ETAMP,ETPHI,EPAMP,EPPHI
      ETT(J,K)=ETT(J,K)+ETTO
      EPP(J,K)=EPP(J,K)+EPPO
  500 CONTINUE
      CALL PRTIM2
C
C
C     IF MORE INTEGRATION GRIDS REMAIN LOOP BACK
C
      IF(I-NG1)600,700,700
  600 CALL RESET(I,MMAX,NMAX)
      GO TO 100
  700 I=I+1
C     ESTABLISH DIRECT RADIATION ON OUTPUT GRID
      WRITE(6,2009)
      DO 750 K=1,KMAX
      TO=PP(K)/0.017453293
      WRITE(6,2011)TO
      DO 750 J=1,JMAX
      CALL FIELD2(J,K,ETTO,EPPO)
      TO=TT(J)/0.017453293
      A1=REAL(ETTO)
      A2=AIMAG(ETTO)
      A3=REAL(EPPO)
      A4=AIMAG(EPPO)
      CALL VECTOR(A1,A2,ETAMP,ETPHI)
      CALL VECTOR(A3,A4,EPAMP,EPPHI)
      WRITE(6,2012)TO,ETAMP,ETPHI,EPAMP,EPPHI
      ETT(J,K)=ETT(J,K)+ETTO
      EPP(J,K)=EPP(J,K)+EPPO
  750 CONTINUE
      CALL PRTIM2
C
C     TRANSLATE PHASE CENTER, SCALE AMPLITUDES, AND OUTPUT TOTAL FIELDS
C
      PUNCH 1001,TITLE
      WRITE(6,2002)TITLE
      WRITE(6,2010)XT,YT,ZT,SCALE
      DO 760 K=1,KMAX
      TO=PP(K)/0.017453293
      WRITE(6,2011)TO
      DO 760 J=1,JMAX
      TO=TT(J)/0.017453293
      A1=REAL(ETT(J,K))
```

Printout D-1 (contd)

```
      A2=AIMAG(ETT(J,K))
      A3=REAL(EPP(J,K))
      A4=AIMAG(EPP(J,K))
      CALL VECTOR(A1,A2,ETAMP,ETPHI)
      CALL VECTOR(A3,A4,EPAMP,EPPHI)
      DP=XT*SITT(J)*COPP(K)+YT*SITT(J)*SIPP(K)+ZT*(COTT(J)+1.0)
      DP=DP*PC*57.29578
      ETPHI=ETPHI-DP
      EPPHI=EPPHI-DP
      CALL ADJUST(ETPHI)
      CALL ADJUST(EPPHI)
      ETAMP=ETAMP*SCALE
      EPAMP=EPAMP*SCALE
      WRITE(6,2012)TO,ETAMP,ETPHI,EPAMP,EPPHI
      PUNCH 2013,TO,ETAMP,ETPHI,EPAMP,EPPHI
  760 CONTINUE
      CALL PRTIM2
C
C     TEST FOR ANOTHER DATA BLOCK
      READ(5,1001)TITLE
      IF(TITLE(1)-LC)1,800,1
  800 CONTINUE
      WRITE(6,2004)
      STOP
 1001 FORMAT(13A6)
 1002 FORMAT(10F10.0)
 2001 FORMAT(32H1       LUDWIG SCATTERING PROGRAM//5X,13A6//)
 2002 FORMAT(5H1     ,13A6//)
 2003 FORMAT(26H      PROPAGATION CONSTANT=,E14.8//)
 2004 FORMAT(18H1 END OF LAST CASE)
 2006 FORMAT(29HO  SCATTERED FIELDS FROM GRID,I2)
 2007 FORMAT(30H1  BEGIN INTEGRATION OVER GRID,I2)
 2009 FORMAT(40H1  DIRECT RADIATION FROM INCIDENT FIELDS)
 2010 FORMAT(63H   SUPERPOSITION OF ALL GRID SCATTERED FIELDS AND DIRECT
     1 FIELDS/32H   PHASE CENTER TRANSLATED BY X=,F10.4,5H   Y=,F10.4,
     25H   Z=,F10.4/
     339H   AMPLITUDE VALUES SCALED BY FACTOR OF,E15.8)
 2011 FORMAT(7HO  PHI=,F7.2/
     117X,7HE THETA,15X,5HE PHI/
     29H    THETA,2(20H     VOLTS    PHASE ))
 2012 FORMAT(F9.2,F11.6,F8.2,F12.6,F8.2)
 2013 FORMAT(F10.2,F10.6,F10.2,F10.6,F10.2)
      END
```

(b) Subroutine FIELDS (spherical-wave expansion)

```
$IBFTC FIELD
      SUBROUTINE FIELDS
C
C     THIS SUBROUTINE EVALUATES A SPHERICAL WAVE EXPANSION TO OBTAIN
C     THE MAGNETIC FIELDS ON A SURFACE S. WAVE COEIFFICIENTS ARE INPUT
C     ON CARDS.  A. LUDWIG 1-14-69
C
      COMMON/GRID1/SIT(15),COT(15),SIP(181),COP(181),T(15),P(181)
      COMMON/GRID2/SITT(181),COTT(181),SIPP(8),COPP(8),TT(181),PP(8)
      DIMENSION F(101),G(100)
      DIMENSION NAME(13)
      DIMENSION A(100,2),B(100,2)
C
C     READ IN WAVE COEIFFICIENTS
C
      READ(5,1001)NAME
      WRITE(6,2001)NAME
      READ(5,1001)NAME
      READ(5,1002)LMAX,MCOMP
      READ(5,1003)(J,A(J,1),A(J,2),B(J,1),B(J,2),L=1,LMAX)
      FMC=MCOMP
      WRITE(6,2003)MCOMP
      WRITE(6,2002)NAME
      WRITE(6,2004)(J,A(J,1),A(J,2),B(J,1),B(J,2),J=1,LMAX)
      RETURN
```

Printout D-1 (contd)

```
C
C       ENTRY POINT FOR MAGNETIC FIELDS AT FINITE R
C
        ENTRY FIELD1(MMAX,NMAX,HR,HT,HP,R)
        DIMENSION R( 15, 91)
        COMPLEX HR( 15, 91),HT( 15, 91),HP( 15, 91)
        IENT=1
        M=0
      1 M=M+1
        SN=SIT(M)
        Z=COT(M)
        TOUT=T(M)
        GO TO 99
C
C       ENTRY POINT FOR ELECTRIC FIELDS AT INFINITE R
C
        ENTRY FIELD2(JO,KO,ETTO,EPPO)
        COMPLEX ETTO,EPPO
        IENT=2
        SN=SITT(JO)
        Z=COTT(JO)
        TOUT=TT(JO)
C
C       ESTABLISH LEGENDRE FUNCTION TABLES FOR M-TH THETA VALUE
C
     99 IF(ABS(SN)-.00001)200,100,100
    100 DO 105 N=1,MCOMP
    105 F(N)=0
        NC=LMAX+1
        CALL LEGEND(NC,MCOMP,Z,F)
        DO 110 N=1,LMAX
        T1=N-MCOMP+1
        T2=N+1
        G(N)=T1*F(N+1)-T2*Z*F(N)
    110 G(N)=G(N)/SN
        DO 115 N=1,LMAX
    115 F(N)=F(N)/SN
C       F(L) IS MULTIPLIED BY FMC LATER
        GO TO (300,500),IENT
C       SPECIAL EQUATIONS FOR TH=0 AND TH=180 DEG
    200 IF(MCOMP-1)210,220,210
    210 DO 215 N=1,LMAX
        F(N)=0
    215 G(N)=0
        GO TO (300,500),IENT
    220 DO 225 N=1,LMAX
        FN=N*(N+1)
        F(N)=FN/2.0
    225 G(N)=FN/2.0
        IF(TOUT-1.57)250,250,230
    230 DO 235 N=1,LMAX,2
        F(N+1)=-F(N+1)
    235 G(N)=-G(N)
    250 GO TO (300,500),IENT
C
C       FOR EACH PHI VALUE THE HANKEL FUNCTIONS ARE EVALUATED AT THE
C       CORRESPONDING VALUE FOR R AND THE WAVE EXPANSION IS SUMMED
C
    300 DO 450 N=1,NMAX
        HRR=0
        HRI=0
        HTR=0
        HTI=0
        HPR=0
        HPI=0
        CALL SPHANK(1,R(M,N),S0R,S0I)
        DO 400 L=1,LMAX
        FL=L
        FN=(FL+1.0)/R(M,N)
        NC=L+1
        CALL SPHANK(NC,R(M,N),S1R,S1I)
```

Printout D-1 (contd)

```
      F1R=-A(L,1)*S1R+A(L,2)*S1I
      F1I=-A(L,1)*S1I-A(L,2)*S1R
      F2R=FN*(A(L,1)*SOI+A(L,2)*SOR)
      F2I=FN*(A(L,2)*SOI-A(L,1)*SOR)
      F3R=-B(L,1)*SOR+B(L,2)*SOI
      F3I=-B(L,1)*SOI-B(L,2)*SOR
      HRR=HRR+F2R*F(L)*FL
      HRI=HRI+F2I*F(L)*FL
      F(L)=F(L)*FMC
      HTR=HTR+F1R*G(L)+F2R*G(L)+F3R*F(L)
      HTI=HTI+F1I*G(L)+F2I*G(L)+F3I*F(L)
      HPR=HPR-F1R*F(L)-F2R*F(L)-F3R*G(L)
      HPI=HPI-F1I*F(L)-F2I*F(L)-F3I*G(L)
      SOR=S1R
      SOI=S1I
  400 CONTINUE
      HRR=HRR*SN*COS(FMC*P(N))
      HRI=HRI*SN*COS(FMC*P(N))
      HR(M,N)=CMPLX(HRR,HRI)
      HTR=HTR*COS(FMC*P(N))
      HTI=HTI*COS(FMC*P(N))
      HT(M,N)=CMPLX(HTR,HTI)
      HPR=HPR*SIN(FMC*P(N))
      HPI=HPI*SIN(FMC*P(N))
      HP(M,N)=CMPLX(HPR,HPI)
  450 CONTINUE
      IF(M-MMAX)1,460,460
  460 RETURN
  500 CONTINUE
      ETR=0.0
      ETI=0
      EPR=0
      EPI=0
      DO 550 L=1,LMAX
      F(L)=F(L)*FMC
      ETR=ETR+A(L,1)*F(L)+B(L,1)*G(L)
      ETI=ETI+A(L,2)*F(L)+B(L,2)*G(L)
      EPR=EPR+A(L,1)*G(L)+B(L,1)*F(L)
      EPI=EPI+A(L,2)*G(L)+B(L,2)*F(L)
  550 CONTINUE
      ETR=ETR*SIN(FMC*PP(KO))
      ETI=ETI*SIN(FMC*PP(KO))
      ETTO=CMPLX(ETR,ETI)
      EPR=EPR*COS(FMC*PP(KO))
      EPI=EPI*COS(FMC*PP(KO))
      EPPO=CMPLX(EPR,EPI)
      RETURN
 1001 FORMAT(13A6)
 1002 FORMAT(5I5)
 1003 FORMAT(I5,2E17.8,2X,2E17.8)
 2001 FORMAT(1H ,62H   FIELD DATA INPUT IN THE FORM OF SPHERICAL WAVE CO
     1EFFICIENTS/5X,13A6)
 2002 FORMAT(5X,13A6///20X,4HA(N),32X,4HB(N)/
     15H     N,7X4HREAL,13X,4HIMAG,15X4HREAL,13X,4HIMAG)
 2003 FORMAT(25H   AZMUITHAL ORDER MCOMP=,I2)
 2004 FORMAT(I5,2E17.8,2X,2E17.8)
      END
```

Printout D-1 (contd)

(c) Subroutine FIELDS (far-field approximation)

```
$IBFTC FIELD
      SUBROUTINE FIELDS
C
C     THIS SUBROUTINE ASSUMES FAR-FIELD BEHAVIOR FOR THE INCIDENT FIELDS
C     TO OBTAIN THE MAGNETIC FIELDS ON A SURFACE S
C     A. LUDWIG MARCH 1969
C
      DIMENSION TITLE(13),PSI(181),E(181),EP(181),H(181),HP(181)
      DIMENSION DUM(2)
      COMPLEX EE,EXPJX
      EQUIVALENCE (EE,DUM(1),CO),(DUM(2),SI)
      COMMON/GRID1/SIT( 15),COT( 15),SIP(181),COP(181),T( 15),P(181)
      COMMON/GRID2/SITT(181),COTT(181),SIPP(  8),COPP(  8),TT(181),PP(8)
      DTR=0.017453293
      WRITE(6,2001)
 2001 FORMAT(52H     ILLUMATION FIELD FROM CIRCULARILY SYMMETRIC FEED//)
C
C     READ IN INPUT FIELDS
      READ(5,1001)TITLE
      READ(5,1002)JINM,J0,JIN,IC1,IC2,MCOMP
      FMC=MCOMP
      READ(5,1003)(PSI(J),E(J),EP(J),H(J),HP(J),J=1,JIN)
C
C     FOR IC1 LESS THAN OR EQUAL TO 0 CONVERT FROM DB TO VOLTS
      IF(IC1)10,10,20
   10 DO 15 J=1,JIN
      E(J)=10.0**(E(J)/20.0)
   15 H(J)=10.0**(H(J)/20.0)
   20 CONTINUE
C
C     FOR IC2 GREATER THAN ZERO NEGLECT FEED PHASE PATTERN
      IF(IC2)40,40,30
   30 DO 35 J=1,JIN
      EP(J)=0
   35 HP(J)=0
   40 CONTINUE
C
C     PRINT OUT FIELD ILLUMINATION
      WRITE(6,2002)TITLE
      WRITE(6,2003)(PSI(J),E(J),EP(J),H(J),HP(J),J=1,JIN)
C
C     CONVERT POLAR ANGLES TO RADIANS
      DO 45 J=1,JIN
   45 PSI(J)=DTR*PSI(J)
C
C     CONVERT TO REAL AND IMAGINARY
      IF(IC2)50,50,60
   50 DO 55 J=1,JIN
      TH=DTR*EP(J)
      EE=EXPJX(TH)
      EP(J)=E(J)*SI
      E(J)=E(J)*CO
      TH=DTR*HP(J)
      EE=EXPJX(TH)
      HP(J)=H(J)*SI
   55 H(J)=H(J)*CO
   60 CONTINUE
C
C     SET FIELDS OUTSIDE OF INPUT RANGE EQUAL TO ZERO
      IF(JINM-JIN-1)72,71,70
   70 PSI(JIN+2)=3.142
      E(JIN+2)=0
      EP(JIN+2)=0
      H(JIN+2)=0
      HP(JIN+2)=0
   71 PSI(JIN+1)=PSI(JIN)+.01
      E(JIN+1)=0
      EP(JIN+1)=0
      H(JIN+1)=0
      HP(JIN+1)=0
```

Printout D-1 (contd)

```
   72 CONTINUE
      RETURN
C
C     ENTER HERE FOR GRID1 FIELDS
C
      ENTRY FIELD1(MMAX,NMAX,HR,HT,HF,R)
      DIMENSION R(15,91)
      COMPLEX HR(15,91),HT(15,91),HF(15,91)
C     LINEARILY INTERPOLATE FIELDS TO INTEGRATION GRID POLAR ANGLES
      J=1
      DO 200 M=1,MMAX
  109 IF(PSI(J)-T(M))110,110,120
  110 J=J+1
      GO TO 109
  120 F=(T(M)-PSI(J-1))/(PSI(J)-PSI(J-1))
      OF=1.0-F
      ETR   =F*E(J)+OF*E(J-1)
      ETI   =F*EP(J)+OF*EP(J-1)
      EPR   =F*H(J)+OF*H(J-1)
  100 EPI   =F*HP(J)+OF*HP(J-1)
      DO 200 N=1,NMAX
      T1=ETR   *SIN(FMC*P(N))
      T2=ETI   *SIN(FMC*P(N))
      T3=EPR   *COS(FMC*P(N))
      T4=EPI   *COS(FMC*P(N))
C     ASSUME FAR FIELD BEHAVIOR HR=0,HP=ET,HT=-EP
      HR(M,N)=(0.0,0.0)
      HT(M,N)=CMPLX(-T3,-T4)
      HF(M,N)=CMPLX(T1,T2)
  200 CONTINUE
      RETURN
C
C     ENTER HERE FOR GRID2 FIELDS
      ENTRY FIELD2(JO,KO,ETTO,EPPO)
      COMPLEX ETTO,EPPO
C
C     LINEARLY INTERPOLATE FIELDS TO OUTPUT GRID POLAR ANGLES
      J=1
      JJ=JO
  509 IF(PSI(J)-TT(JJ))510,510,520
  510 IF(J.EQ.181)GO TO 520
      J=J+1
      GO TO 509
  520 F=(TT(JJ)-PSI(J-1))/(PSI(J)-PSI(J-1))
      OF=1.0-F
      ETR    =F*E(J)+OF*E(J-1)
      ETI    =F*EP(J)+OF*EP(J-1)
      EPR    =F*H(J)+OF*H(J-1)
  500 EPI    =F*HP(J)+OF*HP(J-1)
C
      K=KO
      T1=ETR    *SIN(FMC*PP(K))
      T2=ETI    *SIN(FMC*PP(K))
      T3=EPR    *COS(FMC*PP(K))
      T4=EPI    *COS(FMC*PP(K))
      ETTO=CMPLX(T1,T2)
      EPPO=CMPLX(T3,T4)
      RETURN
 1001 FORMAT(13A6)
 1002 FORMAT(10I5)
 1003 FORMAT(5F10.6)
 2002 FORMAT(13A6/47H      POLAR          E-PLANE              H-PLANE/
     150H      ANGLE      VOLTS       DEG        VOLTS       DEG)
 2003 FORMAT(F10.2,F12.6,F8.2,F13.6,F8.2)
      END
```

Printout D-1 (contd)

(d) Subroutine SURF

```
$IBFTC SURFDK
      SUBROUTINE SURF(MMAX,NMAX,F,FT,FP)
C
C     THIS SUBROUTINE PROVIDES THE MAIN PROGRAM WITH RHO AND ITS
C     PARTIAL DERIVATIVES WITH RESPECT TO THETA AND PHI
C     RHO IS DENOTED BY F IN THIS PROGRAM FOR SOME OBSCURE REASON
C
      DIMENSION TIN(15),A0(15),A1(15),A2(15),DA0(15),DA1(15),DA2(15)
      DIMENSION TITLE(12)
      COMMON/GRID1/SIT(15),COT(15),SIP(181),COP(181),T(15),P(181)
      DIMENSION F(15,91),FT(15,91),FP(15,91)
C
C     READ IN FOURIER COEIFFICIENTS
C
    4 READ(5,1001)TITLE
      WRITE(6,2001)TITLE
      READ(5,1002)(TIN(M),A0(M),A1(M),A2(M),DA0(M),DA1(M),DA2(M),
     *M=1,MMAX)
      WRITE(6,2003)(TIN(M),A0(M),A1(M),A2(M),DA0(M),DA1(M),DA2(M),
     *M=1,MMAX)
C
C     CHECK THAT INPUT DATA AND INTEGRATION GRID AGREE
C
      DO 10 M=1,MMAX
      TCHE=T(M)/0.017453293
      IF(ABS(TCHE-TIN(M))-.000001)10,99,99
   99 WRITE(6,2002)TIN(M),TCHE
   10 CONTINUE
C
C     FILL OUT SURFACE TABLE
C
      DO 50 N=1,NMAX
      DO 50 M=1,MMAX
      COP2P=1.0-2.0*SIP(N)*SIP(N)
      F(M,N)=A0(M)+A1(M)*COP(N)+A2(M)*COP2P
      FT(M,N)=DA0(M)+DA1(M)*COP(N)+DA2(M)*COP2P
      SIP2P=2.0*SIP(N)*COP(N)
      FP(M,N)= -A1(M)*SIP(N)-2.0*A2(M)*SIP2P
   50 CONTINUE
      RETURN
 1001 FORMAT(12A6)
 1002 FORMAT(7F10.4)
 2001 FORMAT(44HO  SURFACE DATA INPUT IN THE FORM OF FOURIER/
     *37H   COEFFICIENTS AND THEIR DERIVATIVES//12A6//
     *44H   THETA   A0(THETA)   A1(THETA)   A2(THETA),
     *37H   DA0/DTHETA  DA1/DTHETA  DA2/DTHETA)
 2002 FORMAT(55H  SURFACE INPUT DATA INCONSISTANT WITH INTEGRATION GRID/
     *14H   INPUT ANGLE,F7.2,18H DEG    GRID ANGLE,F7.2,4H DEG)
 2003 FORMAT(F8.2,6F12.4)
      END
```

(e) Subroutine SETUP

```
$IBFTC SETD
      SUBROUTINE SETUP(NG1,I,JMAX,KMAX,MMAX,NMAX)
C
C     THIS PROGRAM ESTABLISHES THE OUTPUT GRID AND UP TO 5 INTEGRATION
C     GRIDS, AND PRECOMPUTES TRIG FUNCTIONS ON ALL OF THE GRID POINTS
C
      DIMENSION MM( 5),NN(5), T( 15, 5),P(181, 5)
      COMMON/GRID1/SIT( 15),COT( 15),SIP(181),COP(181),TG1( 15),PG1(181)
      COMMON/GRID2/SITT(181),COTT(181),SIPP(  8),COPP(  8),TT(181),PP(8)
      DIMENSION DUM(2)
      COMPLEX E,EXPJX
      EQUIVALENCE (E,DUM(1),CO),(DUM(2),SI)
      DATA L1/O606060606374/,L2/O6060606604774/,L3/O6060606636374/
      DATA L4/O6060604747747/,L5/O3413000000000/
```

Printout D-1 (contd)

```
C
C      GRID VALUES MAY BE READ IN BY INPUTING ONLY -MAX VALUES, OR IF
C      EQUALLY SPACED VALUES ARE DESIRED, PROGRAM WILL COMPUTE VALUES
C      GIVEN A STARTING POINT AND AN INCREMENT
C
C
C      READ IN OUTPUT GRID
       READ(5,1001)JMAX,TT1,DTT
       IF(DTT)10,20,10
   10  DO 15 J=1,JMAX
       FJ=J-1
   15  TT(J)=TT1+FJ*DTT
       GO TO 30
   20  READ(5,1002)(TT(J),J=1,JMAX)
   30  READ(5,1001)KMAX,PP1,DPP
       IF(DPP)40,50,40
   40  DO 45 K=1,KMAX
       FK=K-1
   45  PP(K)=PP1+FK*DPP
       GO TO 60
   50  READ(5,1002)(PP(K),K=1,KMAX)
   60  CONTINUE
C
C      READ IN INTEGRATION GRIDS
       READ(5,1001)NG1
       DO 100 I=1,NG1
       READ(5,1001)MM(I),T1,DT
       MMM=MM(I)
       IF(DT)110,120,110
  110  DO 115 M=1,MMM
       FM=M-1
  115  T(M,I)=T1+FM*DT
       GO TO 130
  120  READ(5,1002)(T(M,I),M=1,MMM)
  130  READ(5,1001)NN(I),P1,DP
       NNN=NN(I)
       IF(DP)140,150,140
  140  DO 145 N=1,NNN
       FN=N-1
  145  P(N,I)=P1+FN*DP
       GO TO 100
  150  READ(5,1002)(P(N,I),N=1,NNN)
  100  CONTINUE
C
C      PRINT OUT GRID DATA
       WRITE(6,2001)
       WRITE(6,2006)(L3,J,L5,TT(J),J=1,JMAX)
       WRITE(6,2002)
       WRITE(6,2006)(L4,K,L5,PP(K),K=1,KMAX)
       WRITE(6,2004)NG1
       DO 155 I=1,NG1
       WRITE(6,2005)I
       MMM=MM(I)
       NNN=NN(I)
       WRITE(6,2006)(L1,M,L5,T(M,I),M=1,MMM)
       WRITE(6,2002)

  155  WRITE(6,2006)(L2,N,L5,P(N,I),N=1,NNN)
C
C      ESTABLISH GRID2 TABLE
       DTR=0.017453293
       DO 200 J=1,JMAX
       TT(J)=DTR*TT(J)
       E=EXPJX(TT(J))
       SITT(J)=SI
  200  COTT(J)=CO
       DO 210 K=1,KMAX
       PP(K)=DTR*PP(K)
       E=EXPJX(PP(K))
       SIPP(K)=SI
  210  COPP(K)=CO
```

Printout D-1 (contd)

```
C
C      ESTABLISH GRID1 TABLES
       I=0
C
C
C      FOR NG1 GREATER THAN 1, RE-ENTER SUBROUTINE HERE FOR NEW GRID1
C
       ENTRY RESET(I,MMAX,NMAX)
       I=I+1
       MMAX=MM(I)
       DO 400 M=1,MMAX
       TG1(M)=DTR*T(M,I)
       E=EXPJX(TG1(M))
       SIT(M)=SI
  400  COT(M)=CO
       NMAX=NN(I)
       DO 410 N=1,NMAX
       PG1(N)=DTR*P(N,I)
       E=EXPJX(PG1(N))
       SIP(N)=SI
  410  COP(N)=CO
       RETURN
 1001  FORMAT(I5,2F10.0)
 1002  FORMAT(8F10.0)
 2001  FORMAT(50H    THE FOLLOWING OUTPUT GRID HAS BEEN ESTABLISHED/
     125H    ALL ANGLES IN DEGREES//)
 2002  FORMAT(//)
 2004  FORMAT(17HO   THE FOLLOWING,I3,40H INTEGRATION GRIDS HAVE BEEN EST
     1ABLISHED/)
 2005  FORMAT(18HO   SEGMENT NUMBER,I3/)
 2006  FORMAT(5(A6,I3,A2,F9.4))
       END
```

(f) Subroutine PATHL

```
$IBFTC PATHLD
       SUBROUTINE PATHL(RHO,J,K,MMAX,NMAX,GAM)
C      THIS SUBROUTINE EVALUATES THE PATH LENGTH FUNCTION GAMMA
       DIMENSION RHO( 15, 91),GAM( 15, 91)
       COMMON/GRID1/SIT( 15),COT( 15),SIP(181),COP(181)
       COMMON/GRID2/SITT(181),COTT(181),SIPP(  8),COPP(  8),TT(181),PP(8)
       DO 10 M=1,MMAX
       T1=SIT(M)*SITT(J)*COPP(K)
       T2=SIT(M)*SITT(J)*SIPP(K)
       T3=COT(M)*COTT(J)-1.0
       DO 10 N=1,NMAX
   10  GAM(M,N)=RHO(M,N)*(T1*COP(N)+T2*SIP(N)+T3)
       RETURN
       END
```

(g) Subroutine FINT

```
$IBFTC FINTD
       SUBROUTINE FINT(X,Y,F,R,MMAX,NMAX,STOT)
C
C      THIS SUBROUTINE NUMERICALLY INTEGRATES
C      THE RADIATION INTEGRAL
C      A. LUDWIG JUNE 1968
C
       DIMENSION X( 15),Y(181),R( 15, 91)
       DIMENSION DUM(2)
       COMPLEX F( 15, 91,3),STOT(3),E,T1,T2,T3,A,B,C,EXPJX,      F12,F23,
     1F14,SUM,TOT
       EQUIVALENCE (E,DUM(1),CO),(DUM(2),SI)
       DO 10 L=1,3
   10  STOT(L)=(0.0,0.0)
       DO 200 N=1,NMAX
       DY=0.5*(Y(N+1)-Y(N))
```

```
      DO 200 M=1,MMAX
      DS=DY*(X(M+1)-X(M))
      R1=R(M+1,N+1)-R(M,N)
      R2=R(M,N+1)-R(M+1,N)
      R3=R(M,N)+R(M+1,N)
      BE=(R1+R2)*0.5
      CE=(R1-R2)*0.5
      AL=(R3-CE)*0.5
      IF(ABS(BE)-0.01)100,100,110
  100 F3I=BE*0.33333333
      F1I=BE*0.5
      F1R=1.0-F1I*F3I
      F3R=0.5-BE*BE/8.0
      GO TO 140
  110 E=EXPJX(BE)
      F1R=SI/BE
      F1I=(1.0-CO)/BE
      F3R=F1R-F1I/BE
      F3I=(F1R-CO)/BE
  140 IF(ABS(CE)-0.01)150,150,160
  150 F4I=CE*0.33333333
      F2I=CE*0.5
      F2R=1.0-F2I*F4I
      F4R=0.5-CE*CE/8.0
      GO TO 170
  160 E=EXPJX(CE)
      F2R=SI/CE
      F2I=(1.0-CO)/CE
      F4R=F2R-F2I/CE
      F4I=(F2R-CO)/CE
  170 E=EXPJX(AL)
      F12=CMPLX(F1R*F2R-F1I*F2I,F1I*F2R+F1R*F2I)
      F23=CMPLX(F2R*F3R-F2I*F3I,F2I*F3R+F2R*F3I)
      F14=CMPLX(F1R*F4R-F1I*F4I,F1I*F4R+F1R*F4I)
      DO 200 L=1,3
      T1=F(M+1,N+1,L)-F(M,N,L)
      T2=F(M,N+1,L)-F(M+1,N,L)
      T3=F(M,N,L)+F(M+1,N,L)
      B=T1+T2
      C=T1-T2
      A=T3-0.5*C
      SUM=A*F12+B*F23+C*F14
      TOT=E*SUM*DS
  200 STOT(L)=STOT(L)+TOT
      RETURN
      END
```

(h) Subroutine PRTIM

```
$IBFTC PRTM13                  PRINT TIMERS
       SUBROUTINE PRTIM1                                          PRTM  20
C      VERSION 2, FEBRUARY 1, 1967                                PRTM  30
C      VERSION 3, APRIL 1968.
       TOTAL=0.                                                   PRTM  40
       PART=0                                                     PRTM  50
       CALL GETTIM (NTIME)                                        PRTM  60
       GO TO 10                                                   PRTM  70
C                                                                 PRTM  80
C                                                                 PRTM  90
C                                                                 PRTM 100
       ENTRY PRTIM2                                               PRTM 110
       CALL GETTIM(ITIME)                                         PRTM 120
       TOTAL=FLOAT(ITIME-NTIME)/3600.                             PRTM 130
       PART=FLOAT(ITIME-IPART)/3600.                              PRTM 140
   10  TOTSEC = TOTAL*60.
       PARSEC = PART*60.
       WRITE(6,1000)PART,PARSEC,TOTAL,TOTSEC
       CALL GETTIM(IPART)                                         PRTM 160
       RETURN                                                     PRTM 170
 1000  FORMAT(15HOPARTIAL TIME =F10.4,5H MINS,3X,1H=,F12.4,5H SECS,
      * 10X,12HTOTAL TIME =F10.4,5H MINS,3X,1H=,F12.4,5H SECS)
       END                                                        PRTM 200
```

Printout D-1 (contd)

```
$IBMAP FSTIME                          FETCH AND STORE LOC 5          FSTI  10
*      FSTIME                          FETCH AND STORE LOC 5          FSTI  20
*                                                                     FSTI  30
*                                      CALL SETTIM(ITIME)             FSTI  40
*                                      CALL GETTIM(ITIME)             FSTI  50
*                                                                     FSTI  60
       ENTRY   SETTIM                                                 FSTI  70
       ENTRY   GETTIM                                                 FSTI  80
SETTIM CLA*    3,4                                                    FSTI  90
       STO     5                                                      FSTI 100
       TRA     1,4                                                    FSTI 110
GETTIM CLA     5                                                      FSTI 120
       STO*    3,4                                                    FSTI 130
       TRA     1,4                                                    FSTI 140
       END                                                            FSTI 150
```

Printout D-2. Computer printout of spherical-wave expansion program

(a) Main program (orthogonality technique)

```
$IBFTC MAINDK
C      SPHERICAL WAVE EXPANSION PROGRAM - ORTHOGONALITY TECHNIQUE
C      THIS PROGRAM DETERMINES THE SPHERICAL WAVE COEIFFICIENTS FOR THE
C      M-TH FOURIER COMPONENT OF A RADIATION PATTERN.(SEE STRATTON P 416)
C      COEIFFICIENTS ARE OBTAINED FROM ORTHOGONALITY INTEGRALS
C      THE EXPANSION MATCHES THE FAR-FIELD INPUT PATTERN WITH THE FAR-
C      FIELD FORM OF SPHERICAL WAVES. ART LUDWIG 12/23/68
C
       DIMENSION NAME(13),NAMEJ(13),NDUM(13)
       DIMENSION PA(60),PB(60)
       DIMENSION T(181),E(181),EP(181),H(181),HP(181)
       DIMENSION F(60,181),G(60,181),PM(61)
       DIMENSION A(181,2),B(181,2)
       DIMENSION ACOE(60,2),BCOE(60,2)
       DIMENSION AOUT(1,2),BOUT(1,2)
       DATA NDUM/13*0606060606060/
       DTR=0.017453293
C
C      READ IN M-TH COMPONENT OF INPUT PATTERN
C
     1 READ(5,1001)NAMEJ
       CALL PRTIM1
       READ(5,1002)MCOMP,NMAX
       READ(5,1001)NAME
       READ(5,1002)JMAX,JO,JIN,IC1,IC2
       READ(5,1003)(T(M),E(M),EP(M),H(M),HP(M),M=1,JIN)
C
C      FOR IC1 LESS THAN OR EQUAL TO 0 CONVERT FROM DB TO VOLTS
       IF(IC1)10,10,20
    10 DO 15 J=1,JIN
       E(J)=10.0**(E(J)/20.0)
    15 H(J)=10.0**(H(J)/20.0)
    20 CONTINUE
C      FOR IC2 GREATER THAN 0 NEGLECT PHASE
       IF(IC2)40,40,30
    30 DO 35 J=1,JIN
       EP(J)=0.
    35 HP(J)=0.
    40 CONTINUE
C      PRINT OUT INPUT PATTERN
       WRITE(6,2001)NAMEJ
       WRITE(6,2006)MCOMP
       WRITE(6,2002)NAME
       WRITE(6,2003)(T(M),E(M),EP(M),H(M),HP(M),M=1,JIN)
C      CONVERT TO RADIANS AND REAL AND IMAGINARY
       DO 45 J=1,JIN
    45 T(J)=DTR*T(J)
       IF(IC2)50,50,60
    50 DO 55 J=1,JIN
       TH=DTR*EP(J)
       EP(J)=E(J)*SIN(TH)
       E(J)=E(J)*COS(TH)
       TH=DTR*HP(J)
       HP(J)=H(J)*SIN(TH)
    55 H(J)=H(J)*COS(TH)
    60 CONTINUE
C
C      ESTABLISH A(THETA)DTHETA AND B(THETA)DTHETA VECTORS
C
       A(1,1)=E(1)*T(1)
       A(1,2)=EP(1)*T(1)
       B(1,1)=H(1)*T(1)
       B(1,2)=HP(1)*T(1)
       DO 65 J=2,JIN
       DTH=T(J)-T(J-1)
       A(J,1)=E(J)*DTH
       A(J,2)=EP(J)*DTH
       B(J,1)=H(J)*DTH
    65 B(J,2)=HP(J)*DTH
       CALL PRTIM2
```

Printout D-2 (contd)

```
C
C      OBTAIN F AND G MATRICES
C
       FMC=MCOMP
       DO 80 J=1,JIN
       Z=COS(T(J))
       DO 85 N=1,MCOMP
    85 PM(N)=0.
       NC=NMAX+1
       CALL LEGEND(NC,MCOMP,Z,PM)
       DO 80 I=1,NMAX
       F(I,J)=PM(I)*FMC
       T1=I-MCOMP+1
       T2=I+1
       G(I,J)=T1*PM(I+1)-T2*Z*PM(I)
    80 CONTINUE
C
C      PERFORM NUMERICAL INTEGRATION BY MATRIX MULTIPLICATION
C
       CALL MULT(JIN,NMAX,2,F,A,ACOE,0,60,181,181,2,60,2)
       CALL MULT(JIN,NMAX,2,G,B,ACOE,1,60,181,181,2,60,2)
       CALL MULT(JIN,NMAX,2,F,B,BCOE,0,60,181,181,2,60,2)
       CALL MULT(JIN,NMAX,2,G,A,BCOE,1,60,181,181,2,60,2)
C
C      NORMALIZE COEIFFICIENTS AND COMPUTE POWERS
C
       PI02Z=(3.1415927/2.0)*0.002655
       PTOT=0.
       DO 95 N=1,NMAX
       PA(N)=0.
       PB(N)=0.
       FACT=1.
       IF(MCOMP)92,92,90
    90 DO 91 M=1,MCOMP
       FF=(N-M+1)*(N+M)
    91 FACT=FACT*FF
    92 FF=2*N+1
       FACT=FACT/FF
       FF=2*N*(N+1)
       FACT=FACT*FF
       K=0
    93 K=K+1
       BT=BCOE(N,K)
       AT=ACOE(N,K)
       BCOE(N,K)=BCOE(N,K)/FACT
       ACOE(N,K)=ACOE(N,K)/FACT
       PA(N)=PA(N)+AT*ACOE(N,K)*PI02Z
       PB(N)=PB(N)+BT*BCOE(N,K)*PI02Z
       IF(K-1)93,93,94
    94 CONTINUE
       PTOT=PTOT+PA(N)+PB(N)
    95 CONTINUE
C
C      OUTPUT COEIFFICIENTS
C
       WRITE(6,2001)NAMEJ
       PUNCH 1001,NAMEJ
       PUNCH 1001,NAME
       PUNCH 2010,NMAX,MCOMP
       WRITE(6,2004)MCOMP
       PSUM=0
       DO 100 J=1,NMAX
       PA(J)=PA(J)/PTOT
       PB(J)=PB(J)/PTOT
       PSUM=PSUM+PA(J)+PB(J)
       PUNCH 2005,  J,ACOE(J,1),ACOE(J,2),BCOE(J,1),BCOE(J,2)
   100 WRITE(6,2005)J,ACOE(J,1),ACOE(J,2),BCOE(J,1),BCOE(J,2),PA(J),PB(J)
      *,PSUM
       WRITE(6,2007)PTOT
       CALL PRTIM2
```

Printout D-2 (contd)

```
C
C       OUTPUT SUMMATION OF SPHERICAL MODES
C
        READ(5,1002)JMAX,JIN
        JO=0
        IC1=1
        IC2=-1
        PUNCH 1001,NAMEJ
        PUNCH 1002,JMAX,JO,JIN,IC1,IC2
        WRITE(6,2008)NAMEJ
        WRITE(6,2002)NDUM
        DT=JMAX-1
        DT=180.0/DT
        J=1
        TH=0
        IF(MCOMP-1)170,180,170
  170   DO 175 N=1,NMAX
        F(1,N)=0
  175   G(1,N)=0
        GO TO 215
  180   DO 190 N=1,NMAX
        FN=N*(N+1)
        F(1,N)=FN/2.0
  190   G(1,N)=FN/2.0
        GO TO 215
  200   FJ=J-1
        TH=FJ*DT
        Z=COS(TH*DTR)
        S=SIN(TH*DTR)
        DO 205 N=1,MCOMP
  205   PM(N)=0
        CALL LEGEND(NC,MCOMP,Z,PM)
        DO 210 N=1,NMAX
        F(1,N)=FMC*PM(N)/S
        T1=N-MCOMP+1
        T2=N+1
        G(1,N)=T1*PM(N+1)-T2*Z*PM(N)
  210   G(1,N)=G(1,N)/S
  215   CALL MULT(NMAX,1,2,F,ACOE,AOUT,0, 60,181, 60,2,1,2)
        CALL MULT(NMAX,1,2,G,BCOE,AOUT,1, 60,181, 60,2,1,2)
        CALL MULT(NMAX,1,2,G,ACOE,BOUT,0, 60,181, 60,2,1,2)
        CALL MULT(NMAX,1,2,F,BCOE,BOUT,1, 60,181, 60,2,1,2)
        CALL VECTOR(AOUT(1,1),AOUT(1,2),EAMP,EPHI)
        CALL VECTOR(BOUT(1,1),BOUT(1,2),HAMP,HPHI)
        PUNCH 2009,  TH,EAMP,EPHI,HAMP,HPHI
        WRITE(6,2003)TH,EAMP,EPHI,HAMP,HPHI
        J=J+1
        IF(J-JIN)200,280,300
  280   IF(J-JMAX)200,285,300
  285   ISIGN=1
        TH=180.
        IF(MCOMP-1)370,380,370
  370   DO 375 N=1,NMAX
        F(1,N)=0
  375   G(1,N)=0
        GO TO 215
  380   DO 290 N=1,NMAX
        ISIGN=-ISIGN
        FN=N*(N+1)*ISIGN
        F(1,N)=-FN/2.0
  290   G(1,N)=FN/2.0
        GO TO 215
  300   CONTINUE
        CALL PRTIM2
        GO TO 1
 1001   FORMAT(13A6)
 1002   FORMAT(10I5)
 1003   FORMAT(5F10.0)
 2001   FORMAT(1H1,13A6)
 2002   FORMAT(1HO,13A6/46H      POLAR          E-PLANE              H-PLANE/
```

Printout D-2 (contd)

```
     150H      ANGLE      VOLTS       DEG        VOLTS       DEG)
 2003 FORMAT(F10.2,F12.6,F8.2,F13.6,F8.2)
 2004 FORMAT(51HO   SPHERICAL WAVE COEFFICIENTS FOR AZMUITHAL ORDER,I2//
     120X,4HA(N),32X,4HB(N),27X,28HFRACTION OF TOTAL MODE POWER/
     25H    N,7X,4HREAL,13X,4HIMAG,15X,4HREAL,13X,4HIMAG,
     313X,7HA MODES,6X,7HB MODES,5X,16HCUMULATIVE TOTAL)
 2005 FORMAT(I5,2E17.8,2X,2E17.8,2X,2E14.5,F14.8)
 2006 FORMAT(49HO  INPUT PATTERN FOR AZMUITHAL COMPONENT OF ORDER,I2)
 2007 FORMAT(1HO,19H   TOTAL MODE POWER,E15.8,6H WATTS)
 2008 FORMAT(1H1,13A6/41H   FAR FIELD SUMMATION OF SPHERICAL MODES)
 2009 FORMAT(F10.2,F10.6,F10.2,F10.6,F10.2)
 2010 FORMAT(2I5)
      END
```

(b) Main program (linear equation technique)

```
$IBFTC MAINDK
C      SPHERICAL WAVE EXPANSION PROGRAM - LINEAR EQUATION TECHNIQUE
C      THIS PROGRAM DETERMINES THE SPHERICAL WAVE COEIFFICIENTS FOR THE
C      M-TH FOURIER COMPONENT OF A RADIATION PATTERN.(SEE STRATTON P 416)
C      THE COEIFFICIENTS ARE OBTAINED BY INVERTING A SYSTEM OF LINEAR
C      EQUATIONS. WARNING- THE INPUT DATA POINTS SHOULD BE NEARLY EQUALLY
C      SPACED BETWEEN 0 AND 180 DEGREES. OTHERWISE THE PROBLEM BECOMES
C      VERY ILL CONDITIONED
C      THE EXPANSION MATCHES THE FAR-FIELD INPUT PATTERN WITH THE FAR-
C      FIELD FORM OF SPHERICAL WAVES. ART LUDWIG 12/23/68
C
       DIMENSION NAME(13)
       DIMENSION T(181),E(181),EP(181),H(181),HP(181)
       DIMENSION A(100,100),XR(100),XI(100),BR(100),BI(100),PM(50)
       DIMENSION C(100,100),S(500)
       DTR=0.017453293
C
C      READ IN M-TH COMPONENT OF INPUT PATTERN
C
     1 READ(5,1001)NAME
       CALL PRTIM1
       READ(5,1002)JMAX,JO,JIN,IC1,IC2,MCOMP
       READ(5,1003)(T(M),E(M),EP(M),H(M),HP(M),M=1,JIN)
C
C      FOR IC1 LESS THAN OR EQUAL TO 0 CONVERT FROM DB TO VOLTS
       IF(IC1)10,10,20
    10 DO 15 J=1,JIN
       E(J)=10.0**(E(J)/20.0)
    15 H(J)=10.0**(H(J)/20.0)
    20 CONTINUE
C      FOR IC2 GREATER THAN 0 NEGLECT PHASE
       IF(IC2)40,40,30
    30 DO 35 J=1,JIN
       EP(J)=0.
    35 HP(J)=0.
    40 CONTINUE
C      PRINT OUT INPUT PATTERN
       WRITE(6,2002)NAME
       WRITE(6,2003)(T(M),E(M),EP(M),H(M),HP(M),M=1,JIN)
C      CONVERT TO RADIANS AND REAL AND IMAGINARY
       DO 45 J=1,JIN
    45 T(J)=DTR*T(J)
       IF(IC2)50,50,60
    50 DO 55 J=1,JIN
       TH=DTR*EP(J)
       EP(J)=E(J)*SIN(TH)
       E(J)=E(J)*COS(TH)
       TH=DTR*HP(J)
       HP(J)=H(J)*SIN(TH)
    55 H(J)=H(J)*COS(TH)
    60 CONTINUE
```

Printout D-2 (contd)

```
C
C      THE COEIFFICIENT VECTOR X IS OBTAINED FROM THE MATRIX EQUATION
C      AX=B, WHERE THE B VECTOR IS DEFINED IN THE FOLLOWING SECTION
C      BR AND BI ARE REAL AND IMAGINARY PARTS OF B
C
       DO 70 J=1,JIN
       BR(2*J-1)=E(J)
       BR(2*J)=H(J)
       BI(2*J-1)=EP(J)
   70  BI(2*J)=HP(J)
C
C      THE ELEMENTS OF THE MATRIX A ARE OBTAINED FROM ASSOCIATED
C      LEGENDRE POLYNOMIALS AND THEIR DERIVATIVES AS FOLLOWS
C
       CALL PRTIM2
       FMC=MCOMP
       DO 80 K=1,JIN
       Z=COS(T(K))
       S=SIN(T(K))
       DO 85 N=1,MCOMP
   85  PM(N)=0.
       NMAX=JIN+1
       CALL LEGEND(NMAX,MCOMP,Z,PM)
       DO 80 J=1,JIN
       F=FMC*PM(J)/S
       T1=J-MCOMP+1
       T2=J+1
       G=T1*PM(J+1)-T2*Z*PM(J)
       G=G/S
       A(2*K-1,2*J-1)=F
       A(2*K,2*J)=F
       A(2*K-1,2*J)=G
       A(2*K,2*J-1)=G
   80  CONTINUE
C
C      THE SYSTEM IS INVERTED TO YIELD X
C
       EPS=.00000001
       NSOLVE=2*JIN
       CALL PRTIM2
       CALL SOLVE(100,A,NSOLVE,BR,XR,1,EPS,2,IT1,C,S)
       CALL SOLVE(100,A,NSOLVE,BI,XI,2,EPS,2,IT2,C,S)
       CALL PRTIM2
       IF(IT1)99,99,90
   90  IF(IT2)99,99,95
   99  WRITE(6,2010) IT1,IT2
   95  CONTINUE
C
C      OUTPUT RESULTS
C
       PUNCH 1001,NAME
       WRITE(6,2004)
       DO 100 J=1,JIN
       PUNCH 2005,  J,XR(2*J-1),XI(2*J-1),XR(2*J),XI(2*J)
  100  WRITE(6,2005)J,XR(2*J-1),XI(2*J-1),XR(2*J),XI(2*J)
       GO TO 1
 1001  FORMAT(13A6)
 1002  FORMAT(10I5)
 1003  FORMAT(5F10.0)
 2002  FORMAT(1H1,13A6/46H      POLAR          E-PLANE              H-PLANE/
     150H      ANGLE      VOLTS       DEG        VOLTS       DEG)
 2003  FORMAT(F10.2,F12.6,F8.2,F13.6,F8.2)
 2004  FORMAT(1H1,29H   SPHERICAL WAVE COEIFFIENTS//
     120X,4HA(N),26X,4HB(N)/
     25H     N,7X,4HREAL,13X,4HIMAG,15X,4HREAL,13X,4HIMAG/)
 2005  FORMAT(I5,2E17.8,2X,2E17.8)
 2010  FORMAT(47H   CAUTION-FULL ACCURACY NOT ACHIEVED IN MATRIX/
     120H      INVERSION. IT1=,I2,7H   IT2=,I2)
       END
```

Printout D-2 (contd)

(c) Subroutine LEGEND

```
$IBFTC LEGDK   FULIST,DECK
       SUBROUTINE LEGEND(NMAX,M,Z,VAL)
C
C      THIS SUBROUTINE RETURNS VALUES OF THE ASSOCIATED LEGENDRE FUNCTION
C      WITH INTEGER INDICES FROM N=M TO N=NMAX.
C      DO NOT USE NMAX LESS THAN M+1
C      VALUES ARE OBTAINED USING UPWARD RECURSION, AND HAVE CHECKED WITH
C      TABULATED VALUES TO 5 PLACES THRU N=56 FOR M=1,AND N=10 FOR M=5
C      HIGHER INDICES WERE NOT CHECKED.
C      ART LUDWIG 6/6/64
       DIMENSION VAL(100)
       DOUBLE PRECISION TERM1,TERM2,TERM3,ZD
       ZD=Z
       FM=M
       TERM1=0.
       IF(M)8,8,9
    8  TERM2=1.0
       GO TO 11
    9  KMAX=2*M-1
       TERM2=(1.0-ZD*ZD)**(FM/2.0)
       DO 10 K=1,KMAX,2
       FK=K
   10  TERM2=TERM2*(2.0*FM-FK)
       VAL(M)=TERM2
   11  NN=NMAX-1
       DO 20 N=M,NN
       FN=N
       TERM3=((2.0*FN+1.0)*ZD*TERM2-(FN+FM)*TERM1)/(FN-FM+1.0)
       VAL(N+1)=TERM3
       TERM1=TERM2
   20  TERM2=TERM3
       RETURN
       END
```

(d) Subroutine SPHANK

```
$IBFTC SPHAND
       SUBROUTINE SPHANK(N,R,X,Y)
C
C      THE VALUE OF THE SPHERICAL HANKEL FUNCTION IS RETURNED, MULTIPLIED
C      BY THE FACTOR (-J)**(N+1) * RHO * EXP(J*RHO)
C
       DOUBLE PRECISION TERM,AR,AI
       AR=0
       AI=0
       PI=3.1415927
       K=0
       TERM=1
       GO TO 100
   20  K=K+1
       T1=N+K
       T2=N-K+1
       T3=2*K
       TERM=TERM*T1*T2/T3
       TERM=TERM/R
       GO TO(200,100),IGO
  100  AR=AR+TERM
       IGO=1
       IF(K-N)20,1000,1000
  200  AI=AI-TERM
       IGO=2
       TERM=-TERM
       IF(K-N)20,1000,1000
 1000  X=AR
       Y=AI
       RETURN
       END
```

Printout D-2 (contd)

(e) Subroutine VECTOR

```
$IBFTC VDECK
      SUBROUTINE VECTOR(X,Y,AMP,PHI)
C     THIS SUBROUTINE CONVERTS COMPLEX DATA TO POLAR FORM
C
      C=0.
      IF(X)100,200,300
  100 IF(Y)110,120,120
  110 C=360.
  120 PHI=ATAN(Y/X)*57.29577951+180.-C
      GO TO 400
  200 IF(Y)210,220,230
  210 PHI=-90.
      GO TO 400
  220 PHI=0.
      GO TO 400
  230 PHI=90.
      GO TO 400
  300 PHI=ATAN(Y/X)*57.29577951
  400 AMP=SQRT(X*X+Y*Y)
      RETURN
      END
```

(f) Subroutine ADJUST

```
$IBFTC ADJUD
      SUBROUTINE ADJUST(PHI)
C     THIS SUBROUTINE SHIFTS PHI UNTIL IT LIES IN THE RANGE -180,180
    1 IF(PHI-180.0)20,20,10
   10 PHI=PHI-360.
      GO TO 1
   20 IF(PHI+180.0)30,40,40
   30 PHI=PHI+360.
      GO TO 20
   40 RETURN
      END
```

(g) Subroutine MULT

```
$IBFTC MULTD   FULIST
      SUBROUTINE MULT(M,N,K,A,B,C,IC,NA,MA,NB,MB,NC,MC)
C     THIS IS A GENERAL PURPOSE MATRIX MULTIPLICATION SUBROUTINE
C
      DIMENSION A(NA,MA),B(NB,MB),C(NC,MC)
      IF(IC)10,20,30
   10 DO 15 I=1,N
      DO 15 J=1,K
      DO 15 L=1,M
   15 C(I,J)=C(I,J)-A(I,L)*B(L,J)
      RETURN
   20 DO 25 I=1,N
      DO 25 J=1,K
      C(I,J)=0.
      DO 25 L=1,M
   25 C(I,J)=C(I,J)+A(I,L)*B(L,J)
      RETURN
   30 DO 35 I=1,N
      DO 35 J=1,K
      DO 35 L=1,M
   35 C(I,J)=C(I,J)+A(I,L)*B(L,J)
      RETURN
      END
```

Printout D-2 (contd)

```
$IBMAP DOTSLV          DOT,SDOT,DAD, ABD ILOG2 FOR USE WITH SOLVE(SLVIT) DOTS  10
*                      17 FEB. 1968                                      DOTS  20
*                       R. J. HANSON, JPL.                               DOTS  30
*   DOT AND FRIENDS     ROUTINES FOR USE WITH SOLVE (SLVIT)              DOTS  40
        ENTRY   DOT (N,A(1),MA,B(1),MB,C)   DOUBLE INNER PRODUCT        DOTS  50
        ENTRY   SDOT (N,A(1),MA,B(1),MB,C)  INNER PRODUCT               DOTS  60
        ENTRY   ILOG2 (A)   FLOATING POINT EXPONENT                     DOTS  70
        ENTRY   DAD (A,B)   ADD WITH ROUND                              DOTS  80
*                                                                       DOTS  90
SNAD    MACRO   M                   STD, COMPLEMENTING IF NECESSARY     DOTS 100
        CHS                                                             DOTS 110
        ADD     =1B2                                                    DOTS 120
        ALS     18                                                      DOTS 130
        STD     M                                                       DOTS 140
        ENDM    SNAD                                                    DOTS 150
SAV     MACRO   A                                                       DOTS 160
        SXA     A,1                                                     DOTS 170
        SXA     A+1,2                                                   DOTS 180
        SXA     A+2,4                                                   DOTS 190
        ENDM    SAV,NOCRS                                               DOTS 200
RETUR   MACRO   A                                                       DOTS 210
        AXT     **,1                                                    DOTS 220
        AXT     **,2                                                    DOTS 230
        AXT     **,4                                                    DOTS 240
        TRA     1,4                                                     DOTS 250
        ENDM    RETUR,NOCRS                                             DOTS 260
DOT     SAV     RET                                                     DOTS 270
        STZ     S                                                       DOTS 280
        STZ     S+1                                                     DOTS 290
        CLA*    8,4                                                     DOTS 300
        LDQ     C+1                                                     DOTS 310
        STO     C                                                       DOTS 320
        CLA*    3,4                                                     DOTS 330
        TZE     NONE                SKIP LOOP IF N = 0                  DOTS 340
        STO     N                                                       DOTS 350
        CLA     4,4                                                     DOTS 360
        PAC     ,1                  X1=-(BASE OF A)                     DOTS 370
        CLA*    5,4                 MA                                  DOTS 380
        SNAD    MA                                                      DOTS 390
        CLA     6,4                                                     DOTS 400
        PAC     ,2                  X2=-(BASE OF B)                     DOTS 410
        CLA*    7,4                                                     DOTS 420
        SNAD    MB                                                      DOTS 430
        LXA     N,4                                                     DOTS 440
 LOOP   LDQ     0,1                 A(I)                                DOTS 450
        FMP     0,2                 B(I)                                DOTS 460
        DFAD    S                                                       DOTS 470
        DST     S                                                       DOTS 480
 MA     TXI     *+1,1,**            (X1)=(X1)+MA                        DOTS 490
 MB     TXI     *+1,2,**            (X2)=(X2)+MB                        DOTS 500
        TIX     LOOP,4,1            END OF MAIN LOOP                    DOTS 510
NONE    DFAD    C                                                       DOTS 520
        FRN                                                             DOTS 530
RET     RETUR                                                           DOTS 540
*                                                                       DOTS 550
SDOT    SAV     SRET                                                    DOTS 560
        STZ     S                                                       DOTS 570
        CLA*    8,4                                                     DOTS 580
        STO     C                                                       DOTS 590
        CLA*    3,4                                                     DOTS 600
        TZE     SNONE                                                   DOTS 610
        STO     N                                                       DOTS 620
        CLA     4,4                                                     DOTS 630
        PAC     ,1                                                      DOTS 640
        CLA*    5,4                                                     DOTS 650
        SNAD    SMA                                                     DOTS 660
        CLA     6,4                                                     DOTS 670
        PAC     ,2                                                      DOTS 680
        CLA*    7,4                                                     DOTS 690
        SNAD    SMB                                                     DOTS 700
```

Printout D-2 (contd)

```
        LXA     N,4                                                     DOTS 710
SLOOP   LDQ     0,1                                                     DOTS 720
        FMP     0,2                                                     DOTS 730
        FAD     S                                                       DOTS 740
        STO     S                                                       DOTS 750
SMA     TXI     *+1,1,**                                                DOTS 760
SMB     TXI     *+1,2,**                                                DOTS 770
        TIX     SLOOP,4,1                                               DOTS 780
SNONE   FAD     C                                                       DOTS 790
        FRN                                                             DOTS 800
SRET    RETUR                                                           DOTS 810
*                                                                       DOTS 820
ILOG2   CAL*    3,4                                                     DOTS 830
        ANA     =0377000000000                                          DOTS 840
        SUB     =0200000000000                                          DOTS 850
        ARS     27                                                      DOTS 860
        TRA     1,4                                                     DOTS 870
*                                                                       DOTS 880
DAD     CLA*    3,4                                                     DOTS 890
        FAD*    4,4                                                     DOTS 900
        FRN                                                             DOTS 910
        TRA     1,4                                                     DOTS 920
*                                                                       DOTS 930
        EVEN                                                            DOTS 940
C       PZE                                                             DOTS 950
        PZE                                                             DOTS 960
S       PZE                                                             DOTS 970
        PZE                                                             DOTS 980
N       PZE                                                             DOTS 990
T       PZE                                                             DOTS1000
        END                                                             DOTS1020
```

(h) Subroutine SOLVE

```
$IBFTC SLVEIT  FULIST,DECK                                              SLVE0010
      SUBROUTINE SOLVE(L,A,N,B,X,IN,EPS,ITMAX,IT,AA,S)                  SLVE0020
CSOLVE   LINEAR EQUATION SOLVER WITH ITERATIVE IMPROVEMENT   VERSION II SLVE0030
C     SOLVES AX=B WHERE A IS NXN MATRIX AND B IS NX1 VECTOR             SLVE0040
C                                                                       SLVE0050
C     IN=                                                               SLVE0060
C         1 FOR FIRST ENTRY                                             SLVE0070
C         2 FOR SUBSEQUENT ENTRIES WITH NEW B                           SLVE0080
C         3 TO RESTORE A AND B                                          SLVE0090
C         4 IF FIRST ENTRY BUT AN INITIAL APPROXIMATION IS ALREADY      SLVE0100
C         AVAILABLE IN X                                                SLVE0110
C         5 CONTINUE CALCULATING ITERATIVE IMPROVEMENT FOR THIS SYSTEM  SLVE0120
C                                                                       SLVE0130
C     EPS AND ITMAX ARE PARAMETERS IN THE ITERATION                     SLVE0140
C     IT=                                                               SLVE0150
C         -1 IF A IS SINGULAR                                           SLVE0160
C         0 IF NOT CONVERGENT                                           SLVE0170
C         NUMBER OF ITERATIONS IF CONVERGENT                            SLVE0180
C     CALLS MAP SUBROUTINES ILOG2, DOT, SDOT AND DAD                    SLVE0190
C                                                                       SLVE0200
      DIMENSION A(L,L),B(L),X(L),AA(N,N),S(1)                           SLVE0210
      MA=L                                                              SLVE0220
C     MA MUST = DECLARED DIMENSION OF SYSTEM                            SLVE0230
      J1=N                                                              SLVE0240
      J2=J1+N                                                           SLVE0250
      J3=J2+N                                                           SLVE0260
      GO TO(10,230,390,10,290),IN                                       SLVE0270
   10 NM1=N-1                                                           SLVE0280
      NP1=N+1                                                           SLVE0290
C                                                                       SLVE0300
C     EQUILIBRATION                                                     SLVE0310
C                                                                       SLVE0320
      DO 40 I=1,N                                                       SLVE0330
         KTOP=ILOG2(A(I,1))                                             SLVE0340
         DO 20 J=2,N                                                    SLVE0350
   20 KTOP=MAXO(KTOP,ILOG2(A(I,J)))                                     SLVE0360
      N2=J2+I                                                           SLVE0370
      S(N2)=2.0**(-KTOP)                                                SLVE0380
         DO 30 J=1,N                                                    SLVE0390
```

Printout D-2 (contd)

```
   30 A(I,J)=A(I,J)*S(N2)                                                  SLVE0400
   40 CONTINUE                                                             SLVE0410
C                                                                          SLVE0420
C     SAVE EQUILIBRATED DATA                                               SLVE0430
C                                                                          SLVE0440
      DO 50 I=1,N                                                          SLVE0450
      DO 50 J=1,N                                                          SLVE0460
   50 AA(I,J)=A(I,J)                                                       SLVE0470
C                                                                          SLVE0480
C     GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING                           SLVE0490
C                                                                          SLVE0500
      DO 170 M=1,NM1                                                       SLVE0510
      N3=J3+M                                                              SLVE0520
      TOP=ABS(A(M,M))                                                      SLVE0530
          IMAX=M                                                           SLVE0540
          DO 70 I=M,N                                                      SLVE0550
      IF(TOP-ABS(A(I,M)))60,70,70                                          SLVE0560
   60 TOP=ABS(A(I,M))                                                      SLVE0570
                IMAX=I                                                     SLVE0580
   70     CONTINUE                                                        SLVE0590
          IF(TOP)90,80,90                                                  SLVE0600
   80     IT=-1                                                           SLVE0610
C         *SINGULAR*                                                       SLVE0620
          RETURN                                                           SLVE0630
   90 S(N3)=IMAX                                                           SLVE0640
  100     IF(IMAX-M)130,130,110                                            SLVE0650
  110     DO 120 J=1,N                                                     SLVE0660
               TEMP=A(M,J)                                                 SLVE0670
               A(M,J)=A(IMAX,J)                                            SLVE0680
  120          A(IMAX,J)=TEMP                                              SLVE0690
  130     MP1=M+1                                                          SLVE0700
          DO 160 I=MP1,N                                                   SLVE0710
               EM=A(I,M)/A(M,M)                                            SLVE0720
               A(I,M)=EM                                                   SLVE0730
               IF(EM)140,160,140                                           SLVE0740
  140          DO 150 J=MP1,N                                              SLVE0750
      IF(ILOG2(A(M,J)) + ILOG2(EM) .LT. -54) GO TO 150                     SLVE0760
               A(I,J)=A(I,J)-A(M,J)*EM                                     SLVE0770
C 150          A(I,J)=A(I,J)-A(M,J)*EM                                     SLVE0780
  150 CONTINUE                                                             SLVE0790
  160     CONTINUE                                                         SLVE0800
  170 CONTINUE                                                             SLVE0810
      N4=N*4                                                               SLVE0820
      S(N4)=N                                                              SLVE0830
      IF (A(N,N))190,180,190                                               SLVE0840
  180 IT=-1                                                                SLVE0850
      RETURN                                                               SLVE0860
  190 CONTINUE                                                             SLVE0870
C     STORAGE FOR A NOW CONTAINS TRIANGULAR L AND U SO THAT (L+I)*U=A      SLVE0880
C                                                                          SLVE0890
C     DUPLICATE INTERCHANGES IN DATA                                       SLVE0900
C                                                                          SLVE0910
      DO 220 I=1,N                                                         SLVE0920
      N3=J3+I                                                              SLVE0930
      IP=S(N3)                                                             SLVE0940
          IF(I-IP)200,220,200                                              SLVE0950
  200     DO 210 J=1,N                                                     SLVE0960
               TEMP=AA(I,J)                                                SLVE0970
               AA(I,J)=AA(IP,J)                                            SLVE0980
  210          AA(IP,J)=TEMP                                               SLVE0990
  220 CONTINUE                                                             SLVE1000
C                                                                          SLVE1010
C     PROCESS RIGHT HAND SIDE                                              SLVE1020
C                                                                          SLVE1030
  230 DO 240 I=1,N                                                         SLVE1040
      N2=J2+I                                                              SLVE1050
  240 B(I)=B(I)*S(N2)                                                      SLVE1060
      DO 250 I=1,NM1                                                       SLVE1070
      N3=J3+I                                                              SLVE1080
      IP=S(N3)                                                             SLVE1090
```

Printout D-2 (contd)

```
          TEMP=B(I)                                                    SLVE1100
          B(I)=B(IP)                                                   SLVE1110
          B(IP)=TEMP                                                   SLVE1120
  250 CONTINUE                                                         SLVE1130
      IF(IN-4) 260,290,290                                             SLVE1140
C                                                                      SLVE1150
C     BYPASS INITIAL APPROXIMATION CALCULATION                         SLVE1160
C     IN GREATER THAN 4 CANNOT OCCUR AT THIS POINT                     SLVE1170
C                                                                      SLVE1180
C                                                                      SLVE1190
C     SOLVE FOR FIRST APPROXIMATION TO X                               SLVE1200
C                                                                      SLVE1210
  260 DO 270 I=1,N                                                     SLVE1220
      N1=J1+I                                                          SLVE1230
  270 S(N1)=-SDOT(I-1,A(I,1),MA,S(J1+1),1,-B(I))                       SLVE1240
      DO 280 K=1,N                                                     SLVE1250
      I=NP1-K                                                          SLVE1260
      N1=J1+I                                                          SLVE1270
  280 X(I)=-SDOT(N-I,A(I,I+1),MA,X(I+1),1,-S(N1))/A(I,I)               SLVE1280
C                                                                      SLVE1290
C     ITERATIVE IMPROVEMENT                                            SLVE1300
C                                                                      SLVE1310
  290 IF(ITMAX)370,370,300                                             SLVE1320
  300 TOP=0.0                                                          SLVE1330
      DO 310 I=1,N                                                     SLVE1340
  310 TOP=AMAX1(TOP,ABS(X(I)))                                         SLVE1350
      EPSX=EPS*TOP                                                     SLVE1360
      DO 360 IT=1,ITMAX                                                SLVE1370
C         FIND RESIDUALS                                               SLVE1380
          DO 320 I=1,N                                                 SLVE1390
  320 S(I)=-DOT(N,AA(I,1), N,X(1),1,-B(I))                             SLVE1400
C         FIND INCREMENT                                               SLVE1410
          DO 330 I=1,N                                                 SLVE1420
      N1=J1+I                                                          SLVE1430
  330 S(N1)=-SDOT(I-1,A(I,1),MA,S(J1+1),1,-S(I))                       SLVE1440
          DO 340 K=1,N                                                 SLVE1450
          I=NP1-K                                                      SLVE1460
      N1=J1+I                                                          SLVE1470
  340 S(I)=-SDOT(N-I,A(I,I+1),MA,S(I+1),1,-S(N1))/A(I,I)               SLVE1480
C         INCREMENT AND TEST CONVERGENCE                               SLVE1490
          TOP=0.0                                                      SLVE1500
          DO 350 I=1,N                                                 SLVE1510
               TEMP=X(I)                                               SLVE1520
      X(I)=DAD(X(I),S(I))                                              SLVE1530
      DELX=ABS(X(I)-TEMP)                                              SLVE1540
      TOP=AMAX1(TOP,DELX)                                              SLVE1550
  350     CONTINUE                                                     SLVE1560
          IF(TOP-EPSX)380,380,360                                      SLVE1570
  360 CONTINUE                                                         SLVE1580
  370 IT=0                                                             SLVE1590
  380 RETURN                                                           SLVE1600
C                                                                      SLVE1610
C     RESTORE A AND B                                                  SLVE1620
C                                                                      SLVE1630
  390 CONTINUE                                                         SLVE1640
      DO 420 K=1,N                                                     SLVE1650
          I=NP1-K                                                      SLVE1660
      N3=J3+I                                                          SLVE1670
      IP=S(N3)                                                         SLVE1680
          IF(I-IP)400,420,400                                          SLVE1690
  400     TEMP=B(I)                                                    SLVE1700
          B(I)=B(IP)                                                   SLVE1710
          B(IP)=TEMP                                                   SLVE1720
          DO 410 J=1,N                                                 SLVE1730
               TEMP=AA(I,J)                                            SLVE1740
               AA(I,J)=AA(IP,J)                                        SLVE1750
  410          AA(IP,J)=TEMP                                           SLVE1760
  420 CONTINUE                                                         SLVE1770
      DO 430 I=1,N                                                     SLVE1780
      N2=J2+I                                                          SLVE1790
```

Printout D-2 (contd)

```
      B(I)=B(I)/S(N2)                                                    SLVE1800
          DO 430 J=1,N                                                   SLVE1810
      A(I,J)=AA(I,J)/S(N2)                                               SLVE1820
  430 CONTINUE                                                           SLVE1830
      RETURN                                                             SLVE1840
      END                                                                SLVE1850
$IBFTC SLVIN.  DECK                                                      SLVI0010
      SUBROUTINE SLVINV (NDIM,N,A,B,IT,S)                                SLVI0020
C                                                                        SLVI0030
C     THE DIMENSION OF S MUST BE AT LEAST N**2+5*N                       SLVI0040
C                                                                        SLVI0050
      DIMENSION B(NDIM,NDIM),S(N)                                        SLVI0060
      ITMAX = 10                                                         SLVI0070
      IN = 1                                                             SLVI0080
      JJ = 4*N                                                           SLVI0090
      DO 20 J = 1,N                                                      SLVI0100
      DO 10 I = 1,N                                                      SLVI0110
      K = JJ + I                                                         SLVI0120
   10 S(K) = 0.0                                                         SLVI0130
      K = JJ + J                                                         SLVI0140
      S(K) = 1.0                                                         SLVI0150
      CALL SOLVE(NDIM,A,N,S(4*N+1),B(1,J),IN,7.0E-9,ITMAX,IT,S(5*N+1),S)SLVI0160
      IF (IT.EQ.(-1)) GO TO 30                                           SLVI0170
      IF (IT.EQ.0) CALL SOLVE (NDIM,A,N,S(4*N+1),B(1,J),5,7.0E-9,ITMAX,  SLVI0180
     1                         IT,S(5*N+1),S)                          SLVI0190
   20  IN = 2                                                            SLVI0200
      CONTINUE                                                           SLVI0210
C                                                                        SLVI0220
C     RESTORE THE MATRIX A                                               SLVI0230
C                                                                        SLVI0240
30    CALL SOLVE(NDIM,A,N,S(4*N+1),B(1,J), 3,7.0E-9,ITMAX,IT,S(5*N+1),S) SLVI0250
      RETURN                                                             SLVI0260
      END                                                                SLVI0270
```

Nomenclature

Coordinates and Related Quantities

x, y, z	Cartesian coordinates (basic coordinate system with the origin located at the source of the incident electromagnetic field)
P	field point (the point at which the scattered or resultant field is evaluated)
$\mathbf{R}$	the vector from the origin to P
R, Θ, Φ	spherical coordinates of P
S	the scattering surface
$\boldsymbol{\rho}$	the vector from the origin to a point on S
ρ, θ, ϕ	spherical coordinates of a point on S
$\rho(\theta, \phi)$	a function describing S
ΔS_{mn}	an incremental area on S
r	the distance from P to a point on S
$\mathbf{n}$	a unit vector normal to S
$\mathbf{i}$	a unit vector in the direction indicated by a coordinate subscript
β	feed offset angle
V	a region of space

Electromagnetic Quantities

$\mathbf{E}$	electric field
$\mathbf{H}$	magnetic field
$\mathbf{E}_i, \mathbf{H}_i$	incident field
$\mathbf{E}_s, \mathbf{H}_s$	scattered field
$\mathbf{E}_T$	total resultant field
H_ρ, H_θ, H_ϕ	components of $\mathbf{H}_i$ with radial variation factored out
E_θ, E_ϕ	components of $\mathbf{E}_i$ with radial variation factored out
E_Θ, E_Φ	components of $\mathbf{E}_s$ with radial variation factored out
$\mathbf{K}$	surface currents
$\Delta\mathbf{K}$	difference between true currents and physical optics currents
$k=\omega(\epsilon\mu)^{1/2}=2\pi/\lambda$	the propagation constant
ϵ	electric permittivity
λ	wavelength
μ	magnetic permeability
ω	angular frequency

Functions and Constants Used in Derivations

$\mathbf{I}$	a vector function related to $\mathbf{E}_s$
$\mathrm{I}_\Theta, \mathrm{I}_\Phi$	components of $\mathbf{I}$
$\Delta\mathbf{I}_{mn}$	incremental contribution to $\mathbf{I}$ from ΔS_{mn}
$a_m(\theta), b_m(\theta)$	Fourier components of surface data
$a'^{e}_{omn}, b'^{e}_{omn}$	coefficients related to the spherical-wave coefficients
$a_n=a^{e}_{omn}$ $b_n=b^{e}_{omn}$	spherical-wave coefficients
$\mathbf{n}_n=\mathbf{n}^{e}_{omn}$ $\mathbf{m}_n=\mathbf{m}^{e}_{omn}$	spherical-wave functions
γ	the path-length function
$\mathbf{F}$	a vector function related to H_i and ρ
$\mathbf{a}_{mn}, \mathbf{b}_{mn}, \mathbf{c}_{mn}$	interpolation coefficients for the function $\mathbf{F}$
$\alpha_{mn}, \beta_{mn}, \xi_{mn}$	interpolation coefficients for the function γ
$A_{m^{e}_{o}}(\theta)=A(\theta)$ $B_{m^{e}_{o}}(\theta)=B(\theta)$	mth even or odd Fourier component of incident electric field

Miscellaneous Symbols

j	$(-1)^{1/2}$
∇	Laplacian operator
$h_n(k\rho)$	spherical Bessel function
$P_n^m(\cos\theta)$	associated Legendre function
f	focal length of a paraboloid
D	physical diameter of source (also used as diameter of a scattering surface)
$F_n^m(\theta), G_n^m(\theta)$	functions related to the associated Legendre function

References

1. Rusch, W. V. T., "Scattering From a Hyperboloidal Reflector in a Cassegrainian Feed System," *IEEE Trans. Ant. Prop.*, Vol. AP-11, No. 4, pp. 414–421, July 1963.

2. Ludwig, A., and Rusch, W. V. T., *Digital Computer Analysis and Design of a Subreflector of Complex Shape*, Technical Report 32-1190. Jet Propulsion Laboratory, Pasadena, Calif., Nov. 15, 1967.

3. Zucker, H., and Ierley, W. H., "Computer-Aided Analysis of Cassegrainian Antennas," The Bell System Technical Journal, pp. 897–932, July–Aug. 1968.

4. Rusch, W. V. T., *Scattering of a Spherical Wave by an Arbitrary Truncated Surface of Revolution*, Technical Report 32-434. Jet Propulsion Laboratory, Pasadena, Calif., May 27, 1963.

5. Ludwig, A., "Computation of Radiation Patterns Involving Numerical Double Integration," *IEEE Trans. Ant. Prop.*, Vol. AP-16, No. 6, pp. 767–769, Nov. 1968.

6. Silver, S., *Microwave Antenna Theory and Design*, Radiation Laboratory Series, Chapter 5, Vol. 12, McGraw-Hill Book Co., Inc., New York, 1949.

7. Tanner, R. L., and Andreasen, M. G., "Numerical Solution of Electromagnetic Problems," *IEEE Spectrum*, pp. 53–61, Sept. 1967.

8. Andreasen, M. G., "Scattering From Bodies of Revolution," *IEEE Trans. Ant. Prop.*, Vol. AP-13, No. 2, pp. 303–310, Mar. 1965.

9. Andreasen, M. G., "Scattering From Parallel Metallic Cylinders With Arbitrary Cross Section," *IEEE Trans. Ant. Prop.*, Vol. AP-12, No. 6, pp. 746–754, Nov. 1964.

10. Cullen, J. A., "Surface Currents Induced by Short-Wavelength Radiation," *Phys. Rev.*, Second Series, Vol. 109, No. 6, pp. 1863–1867, Mar. 15, 1958.

11. Watson, W. H., "The Field Distribution in the Focal Plane of a Paraboloidal Reflector," *IEEE Trans. Ant. Prop.*, Vol. AP-12, No. 5, pp. 561–569, Sept. 1964.

12. King, R., and Wu, T., *The Scattering and Diffraction of Waves*, Harvard University Press, Cambridge, Mass., 1959.

13. Sancer, M., "An Analysis of the Vector Kirchoff Equations and the Associated Boundary-Line Charge," *Radio Science*, Vol. 3, (New Series), No. 2, pp. 141–144, Feb. 1968.

14. Lass, H., *Vector and Tension Analysis*, McGraw-Hill Book Co., Inc., New York, 1950.

15. Dickinson, R. M., *Large Spacecraft Antenna Study Analytical Pattern Subtask*, Technical Report 32-1352. Jet Propulsion Laboratory, Pasadena, Calif., Jan. 15, 1969.

16. Slobin, S. D., *A Study of an Asymmetric Cassegrainian-Antenna Feed System and Its Applications to Millimeter Wave Radio-Astronomical Observations*, USCEE Report 321, Electronic Sciences Laboratory, University of Southern California, Los Angeles, Calif., Dec. 1968.

References (contd)

17. Hogg, D. C., and Semplak, R. A., *An Experimental Study of Near-Field Cassegrainian Antennas,* The Bell System Technical Journal, pp. 2677–2704, Nov. 1964.

18. Ludwig, A. C., "Radiation Pattern Synthesis for Circular Aperture Horn Antennas," *IEEE Trans. Ant. Prop.,* Vol. 14, pp. 434–439, July 1966.

19. Pratt, W. K., and Andrews, H. C., "Computer Calculated Diffraction Patterns," *Applied Optics,* Vol. 7, No. 2, pp. 378–379, Feb. 1968.

20. Allen, C., "Numerical Integration Methods for Antenna Pattern Calculations," *IRE Trans. Ant. Prop.,* Vol. 7, pp. 387–401, Dec. 1959.

21. Rusch, W. V. T., and Strachman, H. L., "Application of a Comprehensive Computer Program to the Analysis of Axisymmetric Microwave Reflector-Antenna Systems," *1968 NEREM Record, IEEE Northeast Electronics Research and Engineering Meeting,* Vol. 10, pp. 20–21, Boston, Nov. 6–8, 1968.

22. Brunstein, S. A., Cormack, R. E., and Ludwig, A. C., "Accuracy of Numerically Computed Electromagnetic Scattered Patterns," in *Supporting Research and Advanced Development,* Space Programs Summary 37-52, Vol. III, pp. 233–238. Jet Propulsion Laboratory, Pasadena, Calif., Aug. 1968.

23. Born, M., and Wolf, E., *Principles of Optics,* Chap. 3, Third Edition. Pergamon Press, Oxford, England, 1965.

24. Davis, P., and Rabinowitz, P., *Numerical Integration,* pp. 64–66, Blaisdell Publishing Co., Waltham, Mass., 1967.

25. *IBM 7094 Principles of Operation,* IBM Systems Reference Library, Form A22-6703-4, IBM Product Publications, pp. 24–36, Poughkeepsie, N.Y., July 1967.

26. Potter, P. D., *The Design of a Very High Power, Very Low Noise Cassegrainian Feed System for a Planetary Radar,* Technical Report 32-653. Jet Propulsion Laboratory, Pasadena, Calif., Aug. 24, 1964.

27. Potter, P. D., "A New Horn Antenna With Suppressed Sidelobes and Equal Beamwidths," *Microwave Journal,* Vol. VI, No. 6, pp. 71–78, June 1963.

28. Levy, G. S., Bathker, D. A., and Ludwig, A. C., "Efficient Antenna Systems: Gain Measurements of the Advanced Antenna System Using *Surveyor I* Signals," in *Supporting Research and Advanced Development,* Space Programs Summary 37-44, Vol. III, pp. 100–105. Jet Propulsion Laboratory, Pasadena, Calif., Jan.–Feb. 1967.

29. Potter, P. D., Merrick, W. D., and Ludwig, A. C., "Big Antenna Systems for Deep Space Communications," *Astronaut. Aeronaut.,* Vol. 4, No. 10, pp. 84–95, Oct. 1966.

30. Ruze, J., "Antenna Cost, Efficiency, and System Noise," *IEEE Trans. Ant. Prop.,* Vol. AP-14, No. 2, pp. 249–250, Mar. 1966.

31. Ruze, J., "Lateral-Feed Displacement in a Paraboloid," *IEEE Trans. Ant. Prop.,* Vol. AP-13, No. 5, pp. 660–665, Sept. 1965.

References (contd)

32. Jones, D. S., *The Theory of Electromagnetism,* Section 8.16, The Macmillan Company, New York, 1964.

33. Galindo, V., "Design of Dual-Reflector Antennas With Arbitrary Phase and Amplitude Distributions," *IEEE Trans. Ant. Prop.,* Vol. AP-12, pp. 403–408, July 1964.

34. Ludwig, A. C., and Norman, R. A., "Correction Factors for Near Field Horn Antenna Gain Measurements," in *Supporting Research and Advanced Development,* Space Programs Summary 37-57, Vol. II, Jet Propulsion Laboratory, Pasadena, Calif., pp. 104–108, May 31, 1969.

35. Stratton, J. A., *Electromagnetic Theory,* Chap. VII, McGraw-Hill Book Co., Inc., New York, 1941.

36. Kennaugh, E. M., *Multiple Field Expansions and Their Use in Approximate Solutions of Electromagnetic Scattering Problems.* Thesis, Ohio State University, Columbus, Ohio, 1959.

37. Papas, C. H., *Theory of Electromagnetic Wave Propagation,* Section 4.4, McGraw-Hill Book Co., Inc., New York, 1965.

38. Bouwkamp, C. J., and Casimir, H. B. G., "On Multiple Expansions in the Theory of Electromagnetic Radiation," *Physica,* Vol. XX, pp. 539–554, 1954.

39. Taylor, T. T., "A Discussion of the Maximum Directivity of an Antenna," *IRE Proceedings,* Vol. 36, p. 1135, Sept. 1948.

40. Harrington, R. F., "Effect of Antenna Size on Gain Bandwidth and Efficiency," *J. Res. NBS, Sec. B,* Vol. 64D, No. 1, pp. 1–12, June 29, 1960.

41. Potter, P. D., "Application of Spherical Wave Theory to Cassegrainian-Fed Paraboloids," *IEEE Trans. Ant. Prop.,* Vol. AP-15, No. 6, pp. 727–736, Nov. 1967.

42. Kennaugh, E. M., and Ott, R. H., *Fields in the Focal Region of a Parabolic Receiving Antenna,* Report 1223-16. Department of Electrical Engineering, Ohio State University Research Foundation, Columbus, Ohio, Aug. 1963.

NASA — JPL — Coml., L.A., Calif.